DOCUMENT RESUME

ED 168 564 IR 007 100

TITLE	Computing In College and University: 1978 and Beyond. Proceedings of the Gerard P. Weeg Memorial Conference, May 1-2, 1978.
INSTITUTION	Iowa Univ., Iowa City. Computer Center.
PUB DATE	May 78
NOTE	143p.; May not reproduce clearly due to small print size
AVAILABLE FROM	G.P. Weeg Computing Center, University of Iowa, Iowa City, IA 52242 ($10.00)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computers; *Computer Science; *Computer Science Education; Educational Technology; Higher Education; *Technological Advancement

ABSTRACT

In most instances, the papers in this collection present information reflecting the current status of computer usage in education and offer substantive forecasts for academic computing. Two speeches from the special ceremony for the renaming of the computing center in honor of Gerard P. Weeg, which was held as part of the two-day national computer conference, are included. Presentation topics include "Dedication: Reflections and Projections," Duane Spriestersbach; "University Response," Willard Boyd; "Computers in Education: Cycles and Trends," John Hamblen; "Equality of Educational Opportunity through Computer Technology," John Eulenberg and Morteza Rahimi; "Pre- and Post-College Computer Education," Arthur Luehrmann; "Future Trends in Computer Science Curricula," Stephen Hedetniemi; "Personal Computing and Its Impact on Education," Seymour Papert; "Computing in the Research Environment," Walter Doherty; "The Case for the Network," Thomas Kurtz; "The Microprocessing Revolution and Its Effect on Small Colleges," Sister Mary Kenneth Keller; "The Reshaping of Men's Minds via Computing," Grace Hertlein; and "Computers, Education, and the Future of Educational Institutions," Alfred Bork. (RAO)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED168564

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

PROCEEDINGS

OF

"COMPUTING IN COLLEGE AND UNIVERSITY: 1978 AND BEYOND"

Gerard P. Weeg Memorial Conference

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
Ted Sjoerdsma

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

May 1-2, 1978
Iowa Memorial Union
University of Iowa
Iowa City, Iowa

IR007100

Copyright © 1978

by

G. P. Weeg Computing Center
University of Iowa

Additional copies may be ordered prepaid from:

WCC Library
Weeg Computing Center
Lindquist Center
University of Iowa
Iowa City, Iowa 52242

The price per copy is $10.00.

Printed in the United States of America

CONFERENCE COMMITTEE

Chairman

Ted Sjoerdsma

Members

Arthur C. Fleck
James W. Johnson
William E. Knabe
W. Lee Shope
Gary A. Wicklund

PROCEEDINGS STAFF

Anne Adair
Lyn Hoversten
Judy Barkalow
Roxanne Berridge
Anne Baustian

TABLE OF CONTENTS

Portrait

Introduction

Dedication: Reflections and Projections
Duane Spriestersbach, University of Iowa..... 1

University Response
Willard Boyd, University of Iowa..... 8

Computers in Education: Cycles and Trends
John Hamblen, University of Missouri-Rolla..... 10

Equality of Educational Opportunity through Computer Technology
John Eulenberg and Morteza Rahimi, Michigan State University.. 26

Pre- and Post-College Computer Education
Arthur Luehrmann, University of California at Berkeley..... 27

Future Trends in Computer Science Curricula
Stephen Hedetniemi, University of Oregon..... 34

Personal Computing and Its Impact on Education
Seymour Papert, Massachusetts Institute of Technology..... 42

Computing in the Research Environment
Walter Doherty, International Business Machines, Inc..... 48

The Case for the Network
Thomas Kurtz, Dartmouth College..... 82

The Microprocessing Revolution and Its Effect on Small Colleges
Sister Mary Kenneth Keller, Clarke College..... 94

The Reshaping of Men's Minds via Computing
Grace Hertlein, California State University at Chico..... 98

Computers, Education, and the Future of Educational Institutions
Alfred Bork, University of California at Irvine.....119

INTRODUCTION

On May 1, 1978, the University Computer Center at the University of Iowa was officially renamed the Gerard P. Weeg Computing Center in a special dedication ceremony held in the Iowa Memorial Union. The event was part of a two-day national computer conference on future directions in academic computing, which addressed many of the same philosophical principles Dr. Weeg espoused until his untimely death on April 10, 1977.

This volume presents the dedication speeches as well as the proceedings from the conference itself as a fitting tribute to a great man.

DEDICATION:
REFLECTIONS AND PROJECTIONS

by

Duane C. Spriestersbach
Vice President for Educational Development and Research
and
Dean of the Graduate College
University of Iowa

It is, of course, a signal honor for me to be asked to make some remarks on this occasion of the dedication of the University Computer Center to the memory of Gerry Weeg. I approached the task with feelings of deep humility and not a little anxiety. How presumptuous of me to attempt to take the measure of a human being for whom I had such great respect and affection, and a human being who worked with such intensity, conviction, and effectiveness in an area that I do not even yet fully comprehend. I make the effort, nevertheless, with the understanding that you will keep in mind my limitations in accomplishing the task.

To set the stage for my remarks, let me review a few of the highlights of Gerry's personal history and accomplishments, even though they may be familiar to many of you. The memorial resolution prepared by Computer Science Professors Don Epley, Art Fleck, and Ted Sjoerdsma, which was read and resolved in a meeting of the faculty of the College of Liberal Arts on April 20, 1977, serves this purpose nicely:

> Gerard P. Weeg was born in Davenport, Iowa, on October 29, 1927. He received his early instruction in Davenport, including his B.S. degree from St. Ambrose College in 1949. A year later while serving as a teaching assistant, he obtained his M.S. degree from Oklahoma State University. He returned to his alma mater to teach mathematics, but after one year obtained a teaching assistantship at Iowa State University, where he completed his Ph.D. degree in mathematics in 1955. He spent two years as a mathematician for

the Remington Rand Univac Corporation in St. Paul, and then from 1956 to 1963 went from Assistant Professor to Professor of Mathematics at Michigan State University. Gerry served as Visiting Professor at Iowa State University for one year prior to accepting his appointment at the University of Iowa, and during a sabbatical in 1969 was Visiting Professor of Computer Science at the University of Massachusetts.

Gerard P. Weeg came to the University of Iowa in 1964 as Director of the University Computer Center and to establish a Computer Science Department within the Division of Mathematical Sciences of the College of Liberal Arts.

Prior to coming to this University, Professor Weeg's academic work was in numerical analysis and algebraic automata theory. His early paper on algebraic automata was widely referenced, and he co-authored a well-received textbook, Introduction to Numerical Analysis.

As a teacher and computer science educator, Professor Weeg's contributions were both universal and personal. Known as "Doc" Weeg to all, with the title signifying affection as well as honor, he attracted many bright young minds into the new discipline of computer science. Indeed, with a handful of other far-sighted educators in the United States, Professor Weeg defined the science of computing and thereby created an academic discipline. As chairman of a new department, Professor Weeg guided the development of a sound academic program with solid academic standards through a difficult period of uncertainty as to which direction the new science should evolve. To his students, he was a superb teacher. His technical expertise and preparation, combined with a natural enthusiasm and flair for humor, made his classes both an education and a delight. To his colleagues, he was the epitome of a good friend: always supportive and concerned, his boundless energy could be immediately turned in their direction by the simplest request.

As Director of the Computer Center, Professor Weeg lead the development of grant proposals which brought funding for major equipment additions to the University and for the development of a true regional center directly serving about fifteen educational institutions in eastern Iowa. He was instrumental in the establishment of a computer-to-computer link between Iowa State University and our own campus. His leadership was strong yet flexible, and his efforts resulted in the potential of academic computing being reached on this campus and others, especially as a service to many other disciplines.

Professor Weeg's ability, expertise, and experience in computing brought recognition throughout the world to him and this University. At the national level, he was one of the founders of a national organization, CONDUIT, whose mission is the development and distribution of computing materials for use in all academic curricula. He helped initiate a now annual national meeting on computing in all aspects of the undergraduate curriculum, familiarly known as CCUC. He served as a visting lecturer for the Association of Computing Machinery, was called upon as a consultant for academic computing by universities and colleges in

seven states, and attended and addressed national and international conferences in Europe and North America. In all of these things, he further distinguished the University by his association with it.

It occurred to me that there was no need to present my abstractions of Gerry's thinking to you. A far better way, it seemed to me, would be to bring you selected examples of his thoughts, pulled from the files that accumulated in my office over the years that he was Director of the Computer Center.

First I have excerpted some statements from some of his more formal writings. This one appeared in the Forward of the Proceedings of a Conference on Computers in the Undergraduate Curricula, held on this campus in June of 1970 with the support of the National Science Foundation:

> It is apparent that the computer presents our society with the possibility of making a division of effort, with those tasks which benefit by man's insight, ingenuity, and instant memory recall to be assigned to man; while those tasks which benefit from the speed, accuracy, repeatability, and indefatigability of an automaton to be assigned to the computer. What is more, the unfolding of computing in government, industry, and research has demonstrated that formerly unsolvable problems become solvable, only to be replaced by facets of those problems, not to mention totally new problems, which reveal themselves as being equally important and more challenging than the solved ones. The analogy to scaling a forbidding peak in a range of mountains, only to discover a new range of even more challenging mountains which were formerly blocked from sight by what once appeared insuperable peaks is, I believe, an appropriate one.
>
> Among the purposes of a college education is discovering and placing in context all those concepts, forces, and events of the past and present whose comprehension ought to allow the individual to weave his way through the complexities of life. It seems clear that among the events and forces of very recent history is the technological revolution, and its central force, the computer. Since it appears to be true that computing is of pervasive importance in our society, every field of collegiate study ought to examine its content <u>vis-a-vis</u> computing. Once again, the mountain range to be revealed in the process, I contend, is one of even greater teaching. The more exciting motivation of working with real data on real problems, with instant reward (or lack of it) for success (or failure), and with the student's assuming full responsibility for his success or failure, all contribute to greater teaching.

In June, 1972, Gerry spoke at the Third Annual Conference on Computers in the Undergraduate Curricula on the topic: "Regional Networks -- Revolution in College Computing." Before he went on to advocate networking, he had this to say about the place of computing in the educational process:

> We want to use the computer in conjunction with classes of every conceivable discipline, from the sciences, to the social sciences, to the humanities and arts. We see the computer as a tool which

can change the dimensions of the ordinary student . . . the student will:

* Solve realistic problems in connection with classes, which can indeed apply to the real world.

* Learn the responsibility which goes with working with a completely accurate but implacable servant, the computer. He will see new levels of problems which were never revealed until mass computation was provided.

* Take pride in work done well with instant self-reward. Computing after all is the ultimate instant reinforcement.

* In short, the student can be motivated to an unusual degree, and motivation is 90% of education.

* The student can stand in the role of a Galileo, Gauss, or Galois, participating in the discovery of general truths by the investigation of specific cases, and verifying that postulated truths are indeed true.

Gerry's thinking about the use of computers in instruction had moved forward at the time that he participated in a panel discussion at the EDUCOM Fall Conference in October, 1973, which was concerned with "Trends in Instructional Use of Computers":

At Iowa, then, we are in the process of rethinking our computer delivery system. For the past three years, we have had collegiate computer fund allocations, which in turn were parcelled out to departments, and from there to individual courses and instructors. Yet, there has been no convenient way to parcel those funds out to the student, for whom, lest we forget, the university exists. The system is fiscally satisfactory, but it in no way contributes to the instructional use of computing.

We have concluded, for a large number of reasons, to back off and start over. Our principal new tenets are:

* Instructional computing implies time-sharing conversational computing.

* Instructional computing should be <u>free</u> to the student.

* A single simple language, BASIC, will handle the bulk of the instructional need.

In March, 1974, Gerry spoke on the "The American Experience" at the Conference on Computers in Higher Education. His conviction about the place of the computer in the educational process continued to be strong:

In my estimation, the role of education is the joining of junior and senior scholars to review and understand the great events, thoughts, forces, and discoveries of the past and present, so that these scholars will be prepared to adapt to, and hopefully control, their future. Clearly, one such force is the computer. I now firmly believe that every college graduate must possess at

least some degree of computer literacy. I believe this can only be attained by the student's having been exposed to the role of computing in most courses which he studies, and not just as a separate subject to be studied apart from his other courses.

And here lies the happy coincidence. For not only can the student thereby acquire computer literacy, he can also learn his subject better, more deeply, with greater relationship to the real world, and probably learn more than by conventional methods. At least this is the conjecture to be proved over the next decade or so.

And now, a couple of excerpts from my files that are less formal, more personal, and on occasion somewhat pungent. On August 1, 1972, he wrote a memo about computing and instruction which included these thoughts:

For the past fifteen years, at least up until the last two or three years, computer center directors have been telling university presidents that though computing is essential to a university, it is cheap. And then directors would proceed to find all sorts of income to support computing, relying on the university itself for only very modest support. The purpose of that charade was to convince administrations that computing could remain on campus. I have also been party to this deception, and only recently has it dawned on me that the disservice was not to the president nor to the director, but to the student and somewhat less to the faculty. For in the process, computer centers become the thralldom of research and commercial activity. Then, when economic stringencies descended upon the nation, they also fell upon computer centers. The result at many schools, and at Iowa, was a strict rationing of computer funds. I am trying to say that faculty morale is going to hell in a handbasket, and if we don't stem that tide, we will lose the 10% of our faculty which has made Iowa a creditable institution.

End of that part of the tirade.

We may be embarking on a phase of university life in which we identify instruction as the most important part of university life. And we have in the last few years seen that instruction become vitalized more and more when computing is used as an adjunct. As a matter of fact, there are those who believe (me) that computing is as integral to instruction as the other great resource, the library.

Now comes my vision. I would like to have computing serve as a spearhead for a re-awakening of the pride and purpose of this university. I would like to see the 10% of our faculty upon whom we pin our hopes for innovation and excellence be rekindled, and then have them bring the rest of the university along. My proposed method is to make computing massively and easily available for all instruction; and to make the availability of such computing so stable and permanent that instructors will spend the long hours needed to rethink and redo their courses. I see that this can be done only by providing a link between the student and the computer of such a nature that the student can maximize the man-machine relationship. In short, I see that it is

imperative for the university to embark on the creation of a mass access interactive computing system.

My dream continues . . . as follows: We need to place large quantities of terminals on this campus. They must be truly available to students, so that I can visualize clumps of ten or more in multiple locations on campus. The computer power to drive more than a handful of these terminals does not exist today at the UCC, and that must be provided. And above all, the use of this system must be free to the student. Indeed, at the student's initial registration at the university, he will receive his student computer ID number which will then allow him full computer use as long as he is a student.

Reprise: In that land flowing with milk and honey which will result from the farsighted decision made now to charge ahead on computing, the faculty will have reacquired its self-confidence; we will have reversed the flow of our good minds back to the university; students will be learning deeply, permanently, and quickly; funding agencies will be clamoring to invest money in our grand enterprise; and the legislature, seeing that we have returned to full vigor, can safely resume beating us about the head and shoulders without fear of our collapsing, thereby depriving it of ways to please the voters.

In a working paper, prepared in October, 1972, entitled "The Realization of the Fourth Revolution at the University of Iowa: A Worm's Eye View", Gerry spelled out his plans in more detail, concluding that "If our objective therefore is to have 50% of our students using the computer for instruction in an intensive fashion, we need . . . 240 terminals minimum. If those students have several computer-oriented courses instead of just one, the number of terminals must increase by some multiplicative factor." This thinking was, of course, at the heart of a proposal which was written almost single-handedly by Gerry but presented to the administration in the name of an ad hoc Committee on Computing under the title of "Computing in Undergraduate Education at the University of Iowa: A Proposal to Make Iowa a Leader in the Use of the Emerging Educational Technology." As many of you know, this report became dubbed the "dream document" which, except for some slippage in the implementation schedule, became and continues to be a blueprint for the integration of computer technology into Iowa's educational programs.

While Gerry was always a forthright person, devoid of deviousness, he was complex. Consequently, more than a few simple phrases are required to characterize him. It has been said of him that he was:

* a truly brilliant person of boundless energy;

* a caring, responsive person who honors human relationships above all;

* a heroic person who dares to take a risk for what he believes is right;

* a person who encouraged, prodded, and counseled in a warm and intimate manner, with frequent reminders to "get on with the show";

* known as "Doc" Weeg to all, with the title signifying affection as well as honor;

* a superb teacher with a natural enthusiasm and a flair for humor;

* a good friend;

* a "whole" person, possessing remarkable professional expertise, and at the same time having the noble human qualities based on high ideals and Christian values.

Gerry's philosophy of life is summarized in a handwritten note that he wrote after he became ill, entitled "Rules that my kids would receive to see them through to a happy old age":

* Be pleasant.

* Be good.

* Be forthright.

* Be upstanding.

* Be not lascivious.

* Be chaste.

* Be holy, which encompasses all of the above.

Gerry was a dreamer and a prophet. But he was more. He was a tenacious advocate for what he believed was right, not unwilling to chide his administrative superiors when they appeared unresponsive or unsympathetic to his dreams. He was also an implementor. As a consequence, he left an extraordinary legacy to the University of Iowa. Each student who sits before a terminal today in conjunction with a course he is taking (there were over 5000 this past academic year) and each member of the faculty who turns to the computer for assistance (there were over 600 of them this academic year) is able to do so, in great measure, because of the vision and effectiveness of Gerry Weeg. We miss his dynamic presence but the impact of it will not cease to grow and must never be forgotten.

UNIVERSITY RESPONSE

by

Willard L. Boyd
President of the University of Iowa

The University of Iowa is forever indebted to Gerry Weeg for his unique contributions to its advancement.

Without question, he has been one of the most significant figures in the life of the University and especially in the period since World War II. While Iowa did not generally expand programs in the 50's and 60's, it did embark into the new and exciting era of the computer. Through Gerry's leadership, Iowa has become a major contributor to computer science and to research and teaching usages of the computer. Through his effort we have a Computer Science Department, perhaps the only marked departure in new departmental subject matter since World War II. As Dean Spriestersbach has indicated, we now are involving each year more and more faculty members and students in using the computer for learning and research. Certainly Gerry's leadership in this major development of University programs has been exceptional.

I remember so many of the times we were together when he pressed steadfastly for better programs for students and faculty. I can remember when we were discussing the old Electrical Engineering Building and East Hall, and we had a short-range and a long-range space solution. There were times when I thought we would never solve the space problem (and we never will), but fortunately, the University Computer Center is now located in facilities worthy of the great program that Gerry built.

I can remember one evening in Old Capitol when Sprie, Gerry and I were arguing with our vendors about the cost of a computer. These have been exciting times for all of us, and Gerry has left an enduring mark on this University.

Gerry's great qualities of intellect, imagination, commitment, and compassion are the qualities which all of us seek. His friendliness and his enthusiasm made every meeting with him not just a discussion, but rather a totally wonderful experience.

He is very much in our thoughts and very much in our lives. This portrait of Gerry will continue to remind us of his contribution to Iowa. On behalf of the University, I am pleased to accept this portrait and to designate our Computer Center as the Gerard P. Weeg Computing Center.

COMPUTERS IN EDUCATION:
CYCLES AND TRENDS

by

John W. Hamblen
Chairman and Professor
Computer Science Department
University of Missouri - Rolla
Rolla, Missouri 65401
(314) 341-4491

Introduction

Extrapolation can be dangerous to your credibility. One of the early warnings given to me as a student in statistics was that extrapolation was extremely risky and should be avoided. This warning notwithstanding, I, along with my colleagues here today, agreed to look into our "crystal balls" and give our best guesses as to the nature of "Computing in Colleges and Universities: 1978 and Beyond". It is too late to change our minds. Our papers are turned in for the proceedings. It is now time for us to unveil our predictions. If Gerry were here now, he would probably say something like "Okay you turkeys, stick your necks way out."

Computer Development

When a new line of computers is introduced, it is done usually with smaller to medium models, and then expanded to larger and even giant systems. This development may extend over a period of years, which appears to be getting shorter with each new generation of computers. We can think of the announcement of a new generation of computers as the beginning of a cycle, and the expansion of the systems as a trend within the cycle. The net picture of computer development is therefore something like Figure 1.

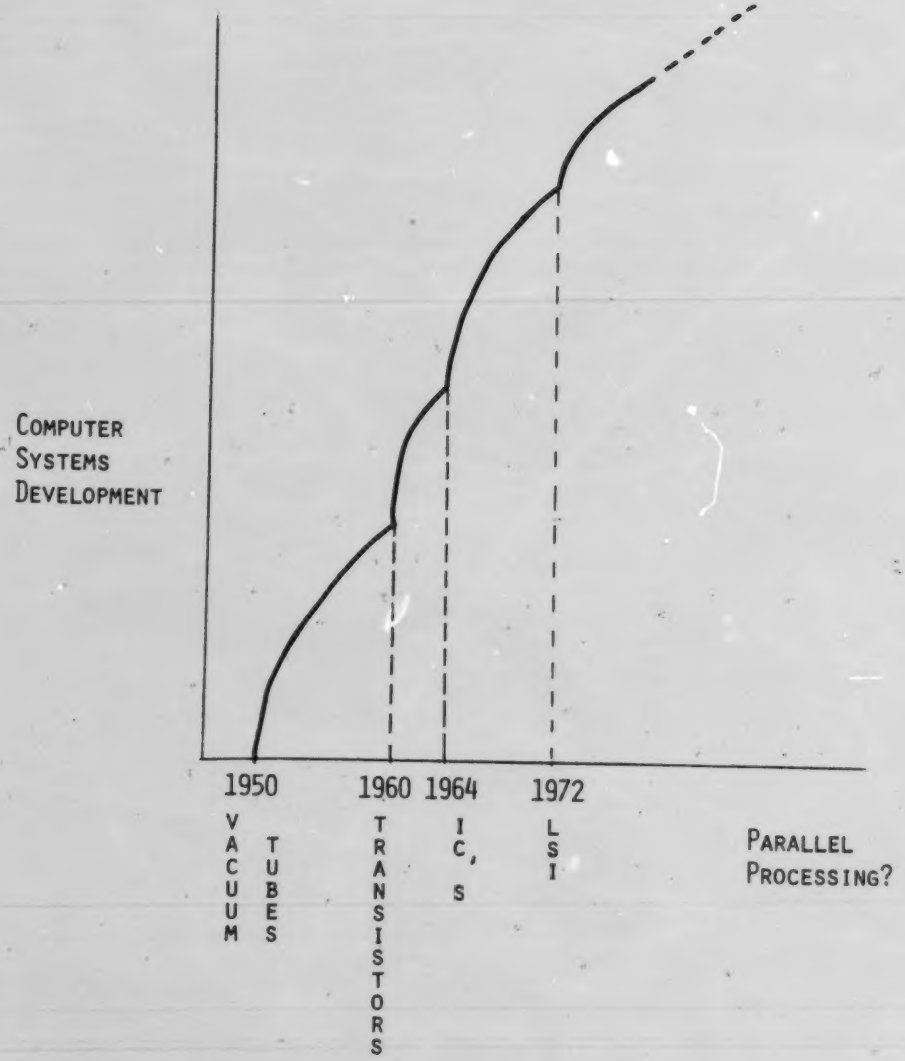

COMPUTER
SYSTEMS
DEVELOPMENT
1950
VACUUM TUBES
1960
TRANSISTORS
1964
IC, S
1972
LSI
PARALLEL
PROCESSING?

FIGURE 1

Each new generation takes us to new heights in speed of computation and density of storage, yet smaller in size and lower in cost per instruction execution. Where will the current LSI cycle take us? Do we need still greater computing speeds? Can we get much more speed out of sequential processing? Or must we look to parallel processing via multiple processors for the ultimate in speed of program execution? The low cost of microprocessors points in this direction, but we have a lot to do before we can manage a bank of processors executing the same program. These problems should keep computer science researchers busy for the next five to ten years. In the meantime, the microcomputers will become as prevalent in five to ten years as the electronic calculator is today.

Expenditures for Computing

If the trend established in the 1960s continues through the '70s, the institutions of higher education will be spending in excess of one billion dollars[1] for computing (see Figure 2). We are constantly reminded that new hardware technologies are giving us more instructions performed per dollar. This point has been driven home to us recently with the introduction of the microcomputer systems. Even today's programmable calculators have more computing power than many of the digital computers that were in use at some colleges and universities during the '50s and early '60s. What, then, is causing computer expenditures to continue to rise? Certainly the quantity of computing being done is growing rapidly. Computers are being used in all disciplines to a certain extent. But this is not the primary cause for rising computing expenditures. It is personnel costs.

Personnel costs are making up a higher percentage of the computing expenditures for several reasons. Increased personnel is required to develop, implement, and operate more complex computer and information systems. Also, the computing field is no longer made up of beginners. On the contrary, there are hundreds of thousands of computing personnel, with ten or more years experience, demanding higher salaries as a result of this experience. Inflation is another major factor. During the last fifteen years, starting salaries for programmers have risen to about 250% of 1963 levels. Part of this increase is due to the higher level of general education and computer education of the entry level personnel. A few short courses from the manufacturer or from a proprietary computer school are no longer sufficient. I see no changes in these trends for the next ten years. The percentage of computing expenditures taken up by personnel costs will continue to rise, perhaps to as high as 75% by 1990 (see Figure 3).

Administrative computing has been taking an increasingly higher percentage of total computing dollars in higher education (see Figure 4). Personnel costs plus increased use are the major contributing factors in the long run, although at present hardware costs are growing also. Distributed systems will become very popular with administrators in the next decade. The financial officers will lead the stampede with their own micro or mini system.

ESTIMATED TOTAL EXPENDITURES FOR COMPUTING

IN U.S. HIGHER EDUCATION THROUGH FY 1980

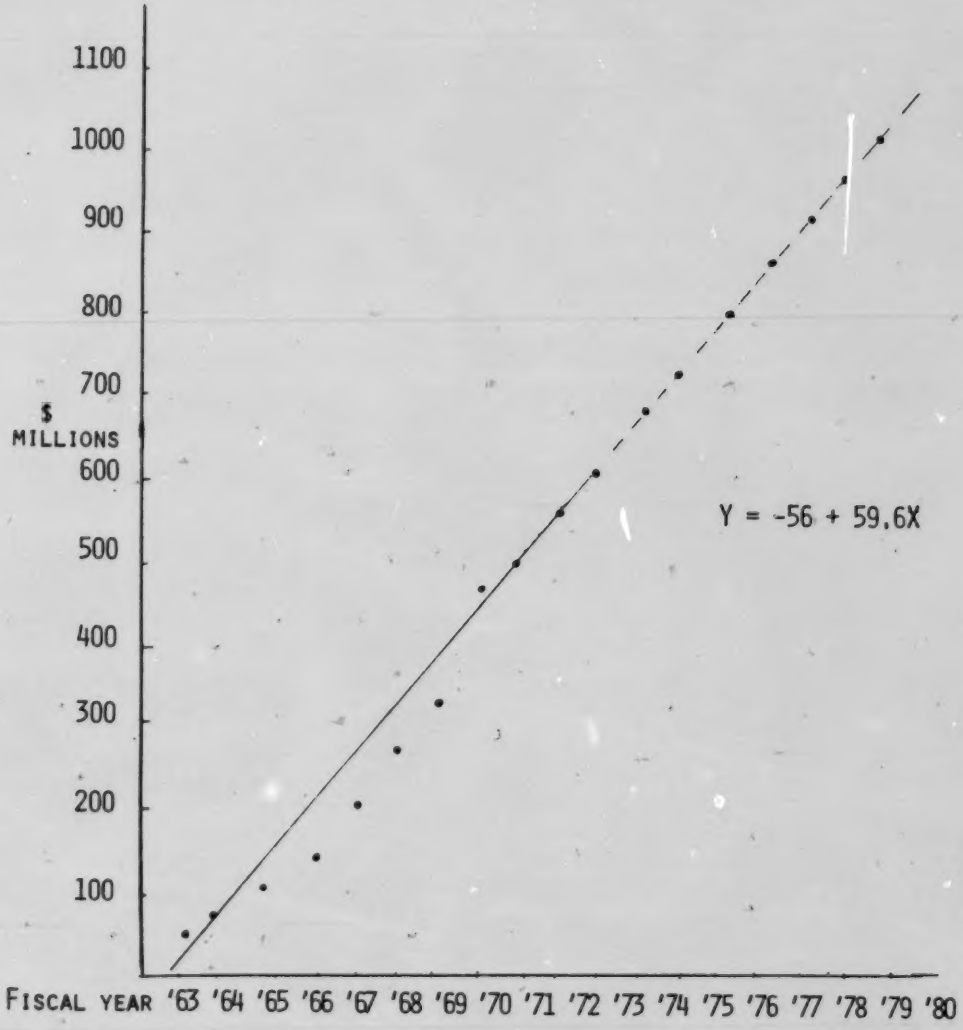
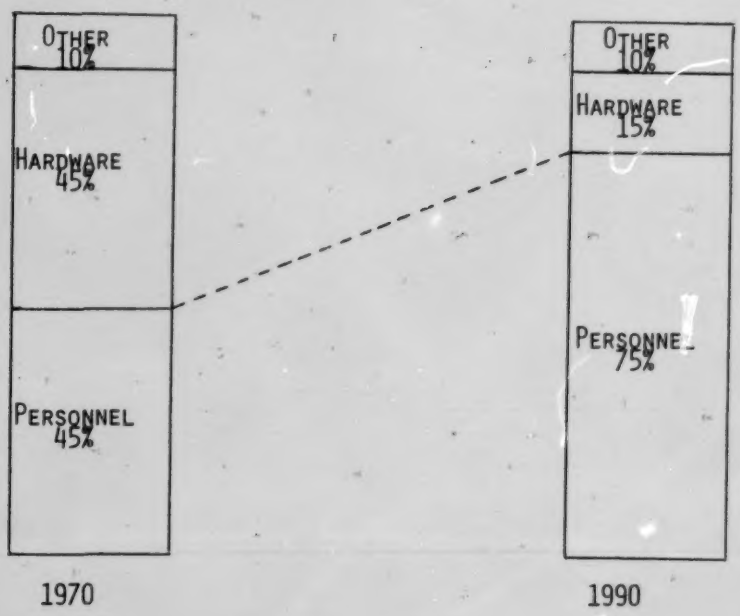


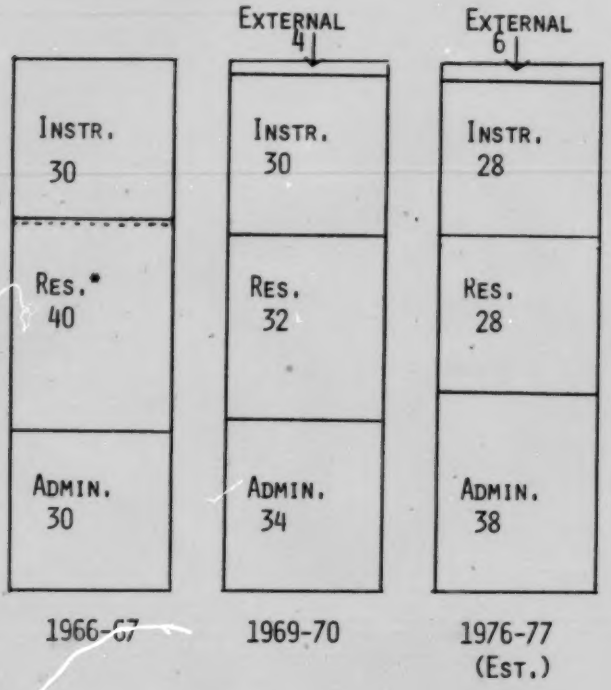
FIGURE 2

FIGURE 3

DISTRIBUTION OF HARDWARE, PERSONNEL, AND OTHER COSTS FOR GENERAL PURPOSE COMPUTING IN HIGHER EDUCATION



ESTIMATED DISTRIBUTION OF EXPENDITURES
FOR
INSTRUCTION, RESEARCH AND ADMINISTRATIVE USES
OF
COMPUTERS IN U.S. HIGHER EDUCATION



*INCLUDES 2%± EXTERNAL

FIGURE 4

Computer Personnel: Supply and Demand

In the second and third references cited at the end of this paper, I have stated that I believe that "many of the problems associated with computer usage is the overutilization of undereducated people." It is not that I am blaming anyone for this situation. We had no choice. A whole new discipline had to be developed to provide the proper education. As computer and information systems became more complex, we found that FORTRAN and BASIC programming skills were not sufficient to cope with them. The logistics of getting the proper and accurate inputs for a large and sophisticated information system challenged the best analysts.

Recommendations have been developed for degree programs in data processing, computer science, and information systems[4-7]. The two-year programs in data processing were thriving in the late '60s; they reached a peak of about 450 programs in 1972. Reductions in federal support and decreased enrollments, partially due to concomitant growth in the number of four-year programs, resulted in many of the weaker two-year programs being closed out. The number of two-year programs has leveled out at about 400 and the number of four-year programs has reached about the same level (see Figure 5).

I do not see much future growth in the number of two-year programs. Furthermore, I believe that those who are responsible for two-year programs cannot afford to ignore the "transfer" student who wants to go on to a four-year program. They should work with the four-year program departments to minimize the loss of credits in the transfer. Catering to this group of students could mean the difference between a weak low-enrollment program and one that is strong and has high enrollments.

Continued growth is expected at the four-year and Masters levels throughout the next decade. Indeed, the 1980 census may show that computer personnel (programmers, analysts, et al) may soon be second only to engineers in the number of persons working in scientific, engineering, and technical occupations.

The colleges and universities have responded fairly well to the computer science curriculum recommendations. However, the business schools have all but ignored them. This is unfortunate, since the need for personnel with such education far exceeds the need for computer scientists. Will the business schools wake up or will it be necessary for the computer science departments to fill the gap? I predict the latter for the next five years, at least.

In the same two references cited above, I published a model I developed for estimating the number of computer personnel positions, by type, existing in 1973 for which employers would prefer to replace with, or place due to growth, personnel having various levels of post-secondary education (see Figure 6).

In the same publications and in the tenth one cited in the back of this paper, I give estimates of computer manpower production at the post-secondary level by states. Figure 7 shows these estimates for two-year programs and above, while Figure 8 compares expected 1977 production with estimated need.

ESTIMATES OF THE NUMBERS OF DEGREE PROGRAMS
IN COMPUTER SCIENCE, DATA PROCESSING, ETC.

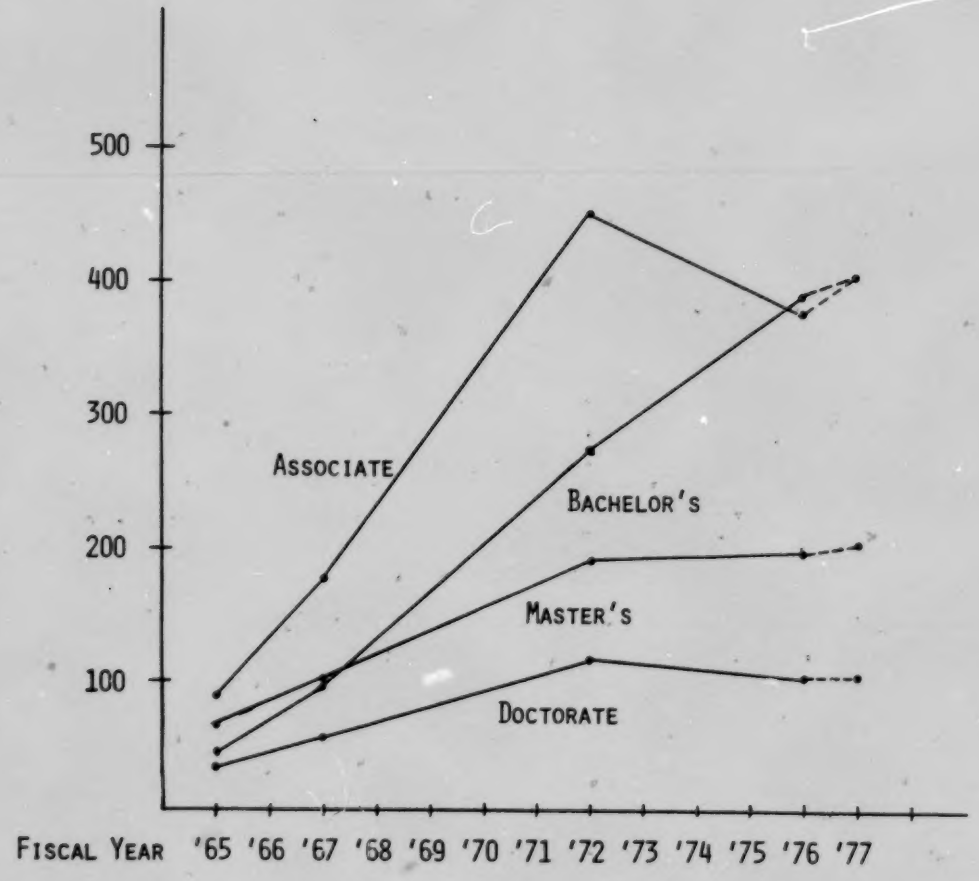
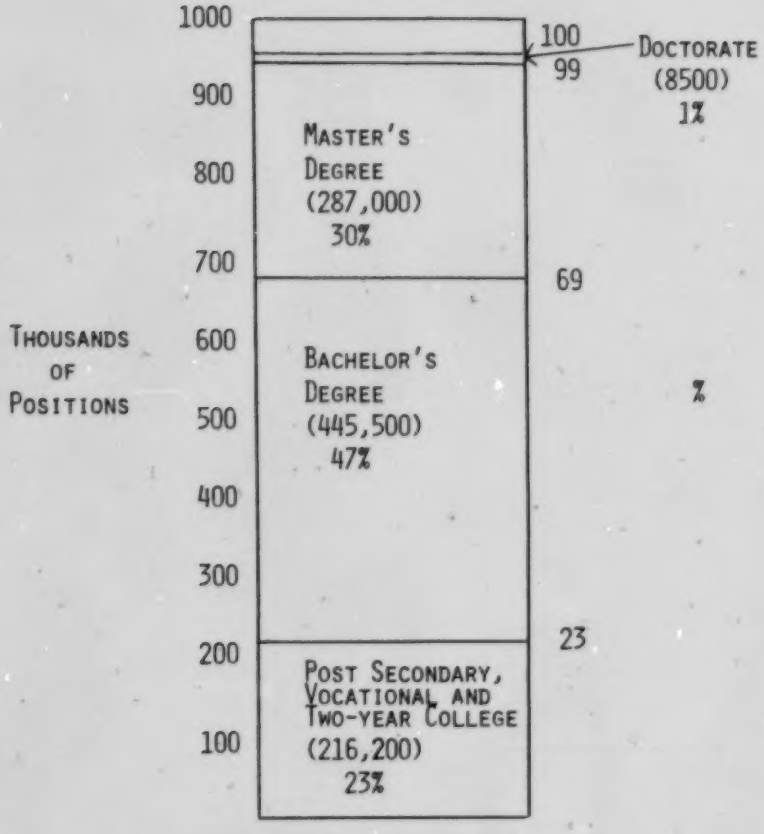


FIGURE 5

DISTRIBUTION OF EXPECTED
TRAINING FOR REPLACEMENTS



BASED UPON 1973 MODEL

FIGURE 6

ESTIMATED NUMBERS OF DEGREES AWARDED

IN COMPUTER SCIENCE, DATA PROCESSING, ETC.

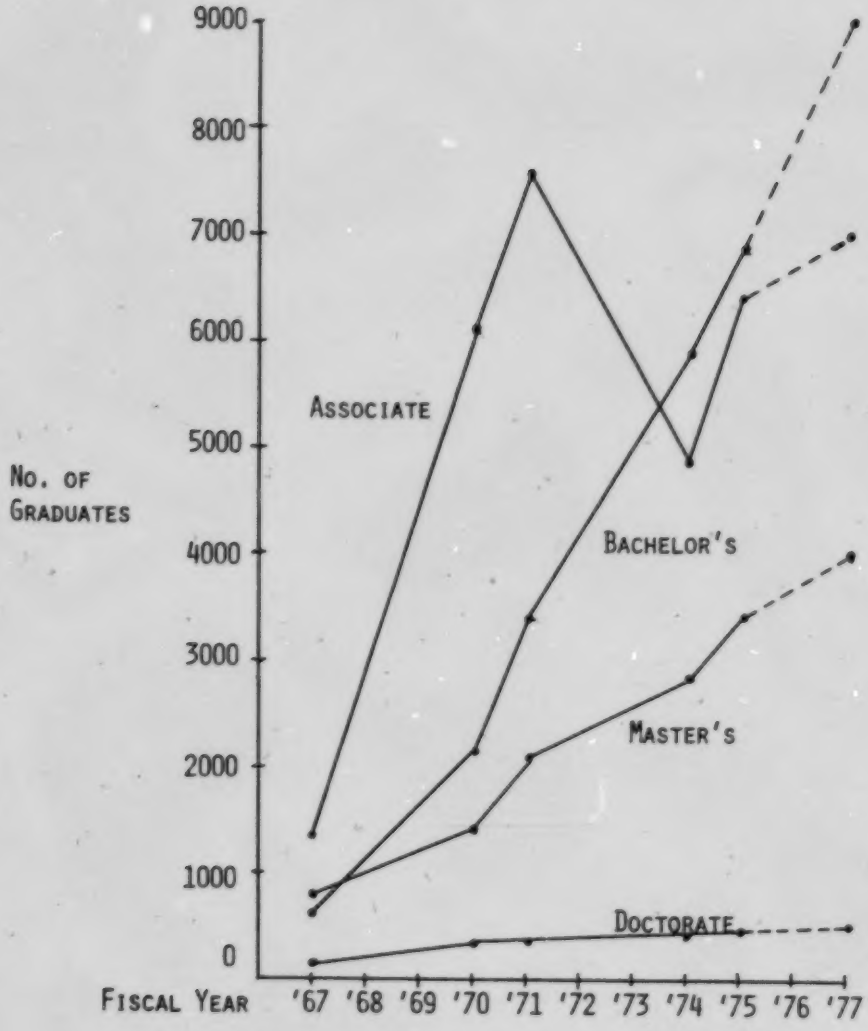


FIGURE 7

ESTIMATED NEW COMPUTER
MANPOWER PRODUCTION
FOR 1977
VS NEEDS

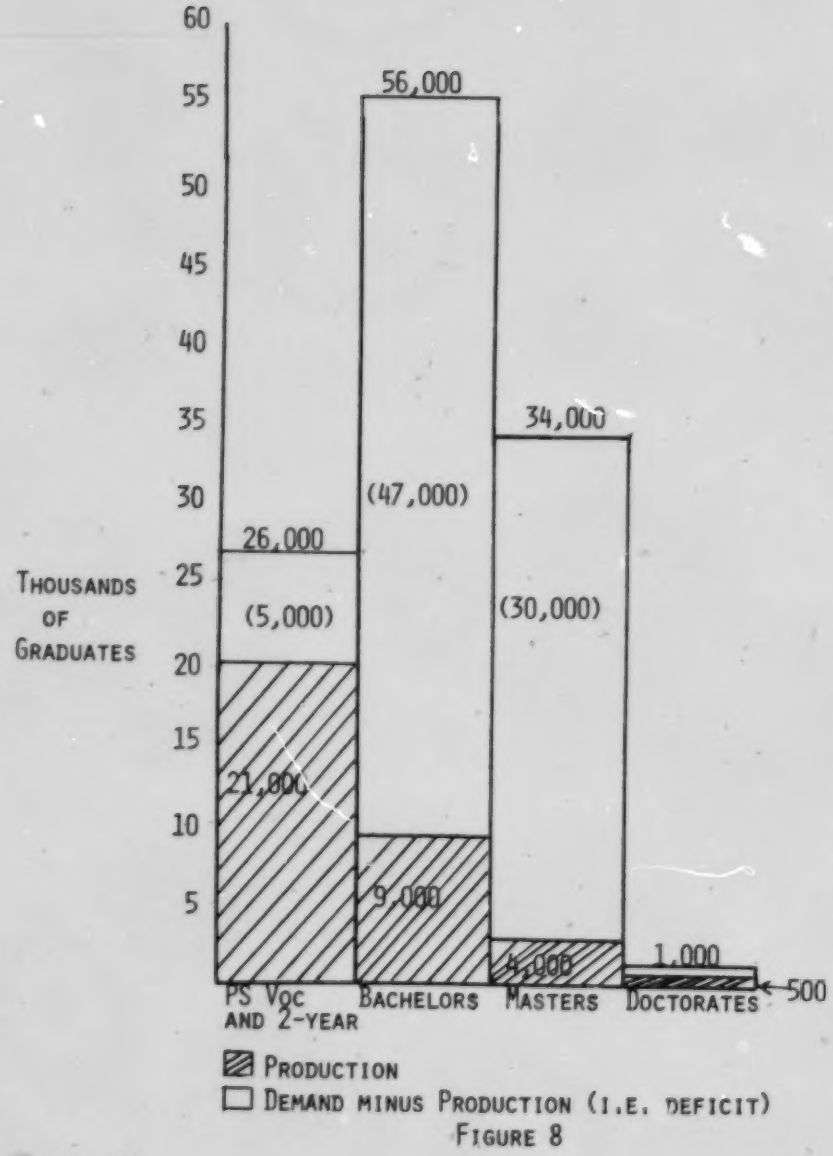


FIGURE 8

page 20

The gaps at the Bachelors and Masters levels are shocking. Even if my need estimates are off by 50% -- and I do not believe they are; maybe 20%, but not 50% -- the gaps would still be shocking. At the 1978 Computer Science Conference's Employment Register in February in Detroit there were about 3000 positions listed, but only 300 applicants!

The Doctorate level supply/demand ratio is much higher, but is very critical, since expansion of the Bachelors and Masters programs is hampered by lack of Doctorate level faculty. Much retreading is still being done and will probably continue for the next five to ten years, as other disciplines are allowed to continue to turn out doctorates who have difficulty in obtaining jobs, and the number of their undergraduate majors continue to decline. Unless some outside factor, such as a massive fellowship program through a government agency (such as the National Science Foundation) is introduced, we are not likely to match supply with demand at the Doctorate level much before 1990.

Computer Education:

"Computer Appreciation", "Computers in Society", and other similar survey courses are currently growing at the college and university level, and will continue to do so for at least another five years. The material in such courses can really be taught at the secondary level, and I believe that it will become popular during the last half of the next decade. How rapidly it spreads is primarily dependent upon how fast the teacher trainers in science and mathematics channel their students into some computer science courses.

The number of institutions offering introductory courses in programming languages will peak during the first half of the 1980s, and then will drop back to 1970 levels by 1990 (see Figure 9). Again this will come about because the high schools will be teaching such courses. Because of the low cost of microcomputers, there is no longer any excuse for high schools not to teach computer programming languages. In fact, I believe that computers in the high school would make high school more interesting, motivating more students to continue their education. By 1990, I expect that courses in BASIC and FORTRAN at the college level will be considered remedial, like courses in trigonometry and college algebra are today. The expected widespread availability of personal computers will help to bring this about.

NUMBERS OF COLLEGES AND UNIVERSITIES OFFERING "COMPUTER APPRECIATION" COURSES WITH ENROLLMENT

	1969-70	1976-77(Est.)	1990
No. Institutions	152*	300	100
Enrollment	18,112	50,000	10,000

*Fifty-six other institutions reported such a course on the books but not yet offered during 1969-70.

COURSES AND ENROLLMENTS IN "INTRODUCTORY COURSES IN SOME COMPUTER LANGUAGE SUCH AS FORTRAN, PL/1, BASIC, ALGOL"

	1969-70	1976-77(Est.)	1990
Courses (Institutions)	971*	2,000	1,000
Enrollment	132,462	300,000	150,000

*Two hundred and six other courses were reported as being on the books but not offered yet during 1969-70.

Figure 9

Computer Uses in Instruction

In instruction, the computer is either regarded as an "end" or a "means". I will say no more about "end" uses, since I am certain that Art Luehrmann and Steve Hedetniemi will cover this thoroughly in their discussions. I do, however, want to make some comments about the future of "means" uses in instruction.

As a "means" in education, the computer may be a delivery system, a tutor, a test administrator, an instructional manager, a tool (e.g., a calculator), a simulator, or a partner in a dialogue (limited). "Means" uses have much in common. Most require interactive capability (i.e., terminals), require relatively slow speeds of transmission, must have low unit costs (i.e., many users required to justify total costs), require massive storage, have common elements which can be shared, and require a large investment in software and courseware development. Therefore, a computer utility or a network is a logical approach to providing the "means" uses of academic computing for all levels of education.

The dedicated commercial timesharing computer utility today, in general, provides dependable service, uses proven software, provides some discount to education, provides nearly continuous access, is still relatively expensive, has users that are quite spread out, and serves many classes of users.

The university-operated computer utility today, in general, provides services only to other educational institutions, provides limited access, faces difficult scheduling problems due to peak loads on their own campuses, is having difficulty expanding facilities to provide needed capacity, provides both remote batch and timesharing services, and has difficulty satisfying customers over a period of time.

The single-node computer utility is now common in higher education. They were encouraged by NSF funding (1968-70), are usually operated by a large university, and provide general purpose computing for education. The quality of their service varies. Off-campus users are usually small colleges and high schools.

In the future, educational computer utilities will be dedicated to certain types of uses. This will be necessary to assure uniformity of instruction, to reduce unit costs, to provide access to instructional materials of uniform quality, to minimize the number of groups of required authors, and to have a uniform delivery system. By serving a cross-section of the education levels, the educational computing utility can attain the needed density of users required to bring unit costs to a practical level by minimizing communications costs and spreading development, maintenance, and operating costs over many users. However, present conditions are not favorable to extensive development of educational computer utilities because of their high initial costs and high communication costs. There is a lack of vehicles for coordination and cooperation among all levels of education to provide the high concentration of users necessary to reduce these costs to a practical level on a per student basis. This is partially due to the fact that a need for computer use at all levels of education is not generally recognized.

To provide the high density of users, coordination is needed between the public school systems and the colleges and universities in a given geographic area. Such coordination can also prevent huge gaps in the utilization of computer technology in education. For the next five to ten years, I believe that the most common successful educational computer utilities will be single-node type, have relatively small CPUs (under $5K/mo), be dedicated to only a few types of services, have users within a 100-mile radius, serve only a few classes of users, use proven software, have users at all levels of education, and provide dependable and nearly continuous access at a total cost of less than $300 per month per terminal (including terminal and communications).

Computer Assisted Instruction (CAI), where the computer presents the instructional material to the student, has had a rocky past. Control Data's gallant efforts to market PLATO is meeting little success. The main reason it is moving slowly is that it is expensive; communications is a significant item in this cost. The other inhibiting factors mentioned above are also at work. Until they are overcome, we will not see general utilization of the computer for CAI in our schools. For some time, I have felt that if we could combine the best of CAI with the best of instructional television, we would have a winner. I am hopeful that the video disk will make this possible by making direct-access segmented video available to CAI systems. I hope that some of the other discussions at this conference will enlighten us on the prospects of video disk technology. It may be that the home entertainment/video game systems industries will take the lead and cause parents to demand that schools utilize this technology.

References

1. J.W. Hamblen, Inventory of Computers in U.S. Higher Education 1969-70: Utilization and Related Degree Programs, Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1972, 401 pg. LC#74-609402 (out of print).

2. J.W. Hamblen, Computer Manpower -- Supply and Demand -- By States (Information System Consultants, St. James, MO, 1973).

3. J.W. Hamblen, "Computer Manpower in the U.S. -- Supply and Demand", SIGSCE Bulletin (December 1975, updated December 1977).

4. J.W. Hamblen, "Using Computers in Higher Education: Past Recommendations, Status, and Needs," Communications of the ACM, 14 (November 1971), 709-712.

5. Curriculum Committee on Computer Education for Management, R.L. Ashenhurst (Editor), "Curriculum Recommendations for Graduate Professional Programs in Information Systems." Communications of the ACM, 16 (May 1972), 363-398.

6. Curriculum Committee on Computer Education for Management, J. Daniel Couger (Editor), "Curriculum Recommendations for Undergraduate Programs in Information Systems", Communications of the ACM, 16 (December 1973), 727-749.

7. Curriculum Committee on Computer Science, Earl Schweppe (Editor) "Curriculum 68 - Recommendations for Academic Programs in Computer Science", Communications of the ACM, 11 (March 1968), 151-197.

8. Curriculum Committee on Computer Science, Richard Austing, et al (Editors), "Curriculum Recommendations for the Undergraduate Program in Computer Science" (A working report), SIGSCE Bulletin (June 1977), 1-16.

9. Committee on Community and Junior College Curriculum, Joyce Currie Little, et al, (Editors), "Curriculum Recommendations and Guidelines for the Community and Junior College Career Program in Computer Programming" (A working report), SIGCSE Bulletin (June 1977), 17-36.

10. J.W. Hamblen, "Computer Education in Higher Education: Status, Alternatives, and Needs", Proceedings of the 1978 National Computer Conference (American Federation of Information Processing Societies, Montvale, NJ, 1978.)

EQUALITY OF EDUCATIONAL OPPORTUNITY THROUGH COMPUTER TECHNOLOGY

by

John Eulenberg and Morteza Rahimi
Department of Computer Science
Michigan State University
East Lansing, Michigan 48823
(517) 355-0831

[Editor's Note: This paper was not available at the time of publication, but a brief description of their presentation has been provided.]

Professors Eulenberg and Rahimi demonstrated the techniques used at Michigan State University to improve the ability of the physically handicapped to enjoy an equality of education through the use of such devices as speech synthesizers, menu boards, specially devised pointers, muscle current sensors, and special-purpose binary switches. All of these devices were driven or enhanced by a digital computer and were used as a means of communication.

PRE- AND POST-COLLEGE COMPUTER EDUCATION

by

Arthur Luehrmann
Lawrence Hall of Science
University of California
Berkeley, California 94720
(415) 642-4193

Introduction

The title and focus of this conference are directed at the role of computing in colleges and universities today and in the future, a role which has been defined and secured by such imaginative and forceful leaders as Gerry Weeg, whom we remember and commemorate today in Iowa City. The other speakers gathered here will attend specifically to the topic, sharing with us their visions of the future of computing in small colleges and research universities. But that future will be shaped in no small part by the extent to which computers become part of the educational landscape in elementary and secondary schools, even in homes. Furthermore, there will exist for several generations a horde of older people who, having failed to learn about computers during their school years, discover as post-graduates that their computer illiteracy is beginning to show.

It may be fitting, therefore, to attend for a while to these two educational environments, the pre- and post-college situations. What computer experiences are students in the future likely to have before they reach college? What educational needs are likely to exist for adult computer illiterates in coming decades? The principal questions that will serve to organize this discussion are these:

1. What computer skills and experiences will kids be picking up before they get to college?

2. What impact will such computer literacy have on colleges?

3. What needs to be done for computer-illiterate adults?

4. What about CAI?

Technology Trends

A great deal of what we all have to say here today is based upon our perceptions of the present and future of computer technology. A brief review is in order.

The underlying phenomenon upon which the so-called "computer revolution" rides is the long-term steep decline in the difficulty (i.e., price) of fabricating larger and larger numbers of circuit elements in smaller and smaller amounts of space and time. This trend, which has steadily held at a 30% per year decrease in the price per element, is independent of the particular technology that has been the favorite (i.e., cheapest) at any given moment. Industry projections into the future, based on known technologies, claim that the 30% per year decline will continue for at least another decade and perhaps two decades. If so, today's prices for computer electronics will be half as much in two years, 1/35 as much in 10 years, and 1/1250 as much in 20 years. Here is another way of expressing the trend: if one agrees that "everyone" can afford to spend $20 for a computer, then in 10 years everyone can have a computer similar to today's Commodore PET or Tandy TRS-80. In 20 years, everyone can have a machine like your favorite $25,000 minicomputer for the same $20.

These remarkable price declines, it should be stressed, apply only to the microelectronic circuits in a computer. Communication costs may decline in the future, but at a far slower rate. Electromechanical systems, such as printers and disc drives, probably will not undergo any significant price change. An obvious consequence of these price trends is the undoing of the old economic argument for connecting many cheap terminals to one expensive processor and giving each user a small portion of the expensive memory. Today a fairly capable personal computer costs less than a dumb terminal. Now and in the future, the components that are so costly that they need to be shared are the electromechanical devices -- high quality printers, discs, and plotters. Economics dictates that what we used to call "peripherals" will be centralized, and the "central computer" will be peripheralized. This "inverted timesharing system" will be held together not by the need to share expensive computer time, but rather to share expensive information in a commonly accessible file system.

Today's stand-alone personal computers are the 4-function electronic calculators of 10 years ago -- cheaper than anything else with the same capabilities, yet too expensive and too limited. They are more responsive than most timesharing terminals, yet too slow for number crunching. An 8K byte memory is enormous, yet too small. Their BASICs are better than one has a right to expect, yet not good enough, not standard, and, and after all, just BASIC. Why not a color display? Floppy discs? Better graphics? Music? Speech?

Indeed, all these facilities are in various stages of appearing. In the next year or two, 8-bit processors will give way to faster 16-bit chips. Memory sizes will rapidly go to a 32K byte "standard". Better operating systems, editors, and language processors will become

available. Yet prices will remain in the $500 to $1000 range as performance variables improve.

What will people do with this richness of computer facility? The editor of a 1977 compendium of articles published in Science magazine on electronics says, "It is clear that the capability of some electronic devices . . . is developing more rapidly than applications can be conceived of." General Motors offers as an $875 Cadillac option the "Trip Computer", an onboard computer with interfaces to the speed, mileage, voltage, engine temperature and fuel supply sensors, and with canned programs to display dynamic information. Mattel and Parker Brothers build computers into strategic games. Atari and Bally build them into kinetic games. When the computer becomes a $20 part, who would buy a TV set without one? Who would buy an electric typewriter without a computer-based editor and several pages of memory? Who would buy a home thermostat that couldn't be set by telephone? Who would buy a telephone that couldn't redial when the busy signal is reached, or forward calls to another phone?

Popular literature these days is full of such examples of the imminent computer invasion. Time magazine, September 5, 1977, quotes a Dallas microcomputer retailer as saying, "Some day soon every home will have a computer. It will be as standard as a toilet." A Wall Street Journal article in April, 1978, quotes an industry estimate that the 1982 U.S. market for personal computers will be $1.8 billion, a five-fold increase from 1978.

The dark cloud in this rosy view of the future will bring us back to our main theme here. As the Time article points out, personal computers "remain complicated devices that require long hours of study to use properly." Computer education is essential if the public is to benefit from the computer revolution. Our enthusiastic Dallas retailer should take note of the fact that one can master the use of the flush toilet just a wee bit faster than one can learn to exploit a computer. How will mass computer education take place?

Pre-College Computer Education

The fact that most of the larger, better known computer education projects have begun in colleges and universities can be misleading. People who already believe that computer use is an arcane skill accessible only to a few specialists with strong mathematics backgrounds are likely to seize on this circumstance as evidence that the college curriculum is the proper place for people to learn to use computers.

Yet the circumstance is an accident of history. Colleges and universities acquired computers at first because of the research needs of the faculty. After the machines were in place, some faculty members found that they could teach their subject matter better if their students were able to use the computer to solve problems or to experiment with simulations or to process data from realistic data bases. Being good faculty types, they communicated with colleagues, wrote papers, and attended topical conferences. We owe the inauguration here at the University of Iowa of the first Conference on Computers in Undergraduate Curricula (CCUC) back in 1970 to Gerry Weeg. The CCUC Proceedings, published every year shortly after the

annual conference is held, give us a history of faculty at the college and university level who are interested in computer uses in their teaching.

A critical look at the topics and concepts presented in CCUC proceedings, however, reveals that a considerable body of this new methodology can be presented successfully to high school, or even junior high school, students. A case in point that I know something about is the application of a computer-based approach to teaching Newtonion mechanics. At the college level, this topic is ordinarily dressed up in the formal language of the calculus, and the fundamental theorem, Newton's Law of Motion, is presented as a differential equation. All physics teachers know that few students are comfortable with this representation, and many students never see past the mathematical dressing far enough to develop an understanding of the dynamic significance of the Law of Motion, which is supposed to predict the motions of all bodies in the universe.

Back in the 1960s, several people -- Alfred Bork, John Robson, a few of us at Dartmouth, and others -- began to experiment with a new mathematical representation of this body of physics. We presented Newton's Law as a finite algorithm for predicting the future position and velocity of a body from a knowledge of its present position and velocity, plus force law which could be an arbitrary function of the present position and velocity. The computer merely exercised this iterative algorithm. Several good things resulted. First, the predictive nature of the law was made manifest. Second, the method was completely general and worked with <u>any</u> force law -- not just the few that permit the differential equation to be solved. Third, the only mathematical background needed was elementary high school algebra. In short, less mathematics was needed to derive a much deeper physical insight.

Now, let us suppose that every high school is equipped with enough computer power to support the use of the above approach in its high school physics course. That being the case, what is the appropriate subject matter for such a high school physics course?

The answer, I believe, is evident. Newton's Law of Motion becomes mathematically accessible to high school students, and there is no need to delay until college the proper treatment of so fundamental a concept. High school students should be taught to represent it as a computer algorithm and should conduct numerous computer-based experiments with different laws of force, different numbers of spatial dimensions, and different numbers of interacting particles. They should emerge with a richer experience and a deeper understanding of the dynamic nature of Newtonian mechanics than most college students get today. At the same time, the mathematical difficulty should seem to be a great deal less than for some of the topics now being taught in high school physics.

New educational opportunities within existing subjects, then, will be a strong driving force for introducing computers into the secondary school curriculum. Even today the process is well under way. A 1978 Minnesota Educational Computer Consortium survey of all secondary school math, science, and business teachers in the state showed that one third of the respondents were using the computer in some of their classes.

The next step will be the creation of new courses specifically to teach ordinary students, not just the top 5-10% who take math and science, how to use computers for a variety of tasks, both numeric and non-numeric. These "basic computing skills" courses will teach about algorithms, writing and debugging programs, running simulation models, gaming, editing text, and drawing graphs and pictures. Unlike previous courses in which one mainly reads about computers, these new ones will require many hours of hands-on experience with computers -- perhaps two to three hours per week per student. Cheap personal computers make this type of course as practical as a typing course.

Such new courses will require teacher education and curriculum development. In-service training of about 30,000 teachers is needed to put one full-time computer teacher in each secondary school. Curriculum development projects are needed to produce materials the teachers and students will use. Both teacher training and curriculum development are appropriate activities for federal support, since the need for basic computing skills is a national one and no other organization will undertake the task. The local community, on the other hand, can be expected to provide ongoing support for computer education, if only because of the growing vocational demand for people with basic skills in computing.

Impact on Colleges

If the above scenario is correct, then we may look forward to steadily increasing computer literacy among high school graduates and college matriculants during the next decade. What impact will this new knowledge have on colleges and universities?

In the first place, students coming to college will expect to find educational computing facilities at least as good as the ones they left behind at high school -- or even at home! Many students today are disappointed to find poorer facilities in college, and there is evidence that the quality and accessibility of the campus computer environment is a significant factor for some students in choosing a college. Prestigious research-oriented institutions often are the losers in this contest, since their above-average computer facilities are usually off-limits to undergraduates.

The college curriculum must adapt also. As indicated earlier, many curricular uses of computers identified in CCUC Proceedings will have gravitated into the high school curriculum. Programming syntax and introductory computer science topics will certainly have been learned there. Consequently, colleges and universities must guard against vain repetition and be prepared to present advanced topics to freshman.

Finally, a great deal of the methodology taught in college-level courses needs to be updated to include or, in a few cases, be replaced by computer-based methods. Digital instrumentation and computer interfacing techniques need to be taught in laboratory science courses. Computer modeling has to be included among other problem-solving methods in applied mathematics and science courses. Modern data retrieval, analysis, and display methods have to be included in social science courses. Linguists need to know how to process text for syntactic and stylistic analysis. Artists and musicians need to know how to make creative use of computers.

Fortunately, as is usually the case with curricular change, each generation of better educated college entrants will carry the message that they already know some of the things we are teaching and that perhaps we should get on to more interesting topics.

Computer-Illiterate Adults

The suddenness with which computer literacy seems likely to be added to the three R's in the school curriculum means that millions of mature well-placed adults face a life of computer illiteracy. No great calamity, one might say; after all, they've done well enough so far. Nevertheless, certain stresses are predictable.

As institutions come to depend more and more on computer-based information systems, a new break of talent is needed. Who will supervise the computer experts? How effective will the supervision and leadership be, if those at the top really don't understand what the computer people do? This is a problem today and will get worse for a long time.

Office automation is just beginning today. The skills needed to perform clerical and typing functions in such an environment are very different from the conventional ones. An office worker needs to be able to work interactively with conversational machines, instruct them to carry out specified tasks, store and retrieve text files, and run diagnostic procedures. What will happen to the job mobility of an older person lacking these skills?

More and more, society will depend upon abilities to program computers, interact with conversational programs, diagnose software and hardware problems, and make changes and repairs to each. Those who have these skills will be in great demand. Others will have a powerful incentive to acquire them.

Finally, many adult computer illiterates who are not threatened by job inadequacy or insecurity will be interested in learning about computers out of natural curiosity, the desire for self-improvement, or a wish to stay current. Such people will bring a computer home just to learn about it.

All of these adults, taken collectively, represent a vast number of people with a need for education of a rather specific kind. During a period of declining enrollments, this ought to be welcome news indeed. The problem, however, is that our institutions of formal education have been notably unsuccessful in serving adults. Unless they develop new approaches, calendars, schedules, and organizational entities suited to this clientele, it will go elsewhere to satisfy its needs, perhaps to proprietary schools, libraries, or science museums.

What about CAI?

I have not thus far painted a bright future for computer assisted instruction, either in computer skills or other subjects. In part, this is because, in my judgement, the greatest educational need regarding computers is a knowledge of their use. This will be the

case whether or not computers can deliver instruction in other subjects more effectively or at a lower cost.

Though it is off the main point of my talk, I venture to guess that CAI will not be a great success within the school system but that it may have much to offer in the non-school environment. CAI is basically incompatible with the school system. Since it offers to do what teachers do now, its adoption would mean the dismissal of teachers. Hooray! say some CAI advocates, since teachers are expensive; CAI, they argue, will lead to a productivity revolution in education similar to that in agriculture and industry. And this would indeed be the case if the laws of economics alone were to dictate the outcome. In that case, inefficient schools would produce a costly product which few would buy. Instead they would take their schooling business to a less costly school using labor-saving methods.

Schools are not like that. They are social and political entities, more than economic ones. Parents lack realistic means of choosing cheaper schools by shopping around on a year-to-year basis. The fundamental market mechanism is absent.

On the other hand, non-school environments (such as the library, museum, and home) are well suited to the use of stand-alone teaching systems, and CAI is likely to find good uses there. No teachers will be threatened, and consumers will shop for the best buys. CAI developers, furthermore, will find there for the first time the mass market needed to justify the large capital investment required to create CAI materials.

Summary

Educational computing got its start in colleges and universities like the one we are meeting at today. The essential ingredients were an imaginative faculty and an accessible computer system. Today (and to a greater extent in the future) inexpensive microcomputers are bringing access to everyone, both inside and outside universities. Many of the curricular innovations that come from universities will gravitate naturally into the high schools, where a new breed of computer teacher will conduct hands-on classes in basic computing skills. College and university matriculants, therefore, will arrive with a solid understanding of the fundamentals, and they will expect to learn more advanced methods and applications. They will also expect to have even greater access to computers than they had in high school. As for the adults left behind in computer illiteracy, new organizations and new methods of instruction will be needed to serve their needs.

FUTURE TRENDS IN COMPUTER SCIENCE CURRICULA

by

Stephen T. Hedetniemi
Department of Computer Science
University of Oregon
Eugene, Oregon 97403
(503) 686-4394

Preface

Let me state at the outset what I hope to accomplish with this statement about future trends in computer science curricula. It is my intention to be as provocative, and yet constructive, as possible; to stir up controversy supported by facts and figures; and to raise a variety of serious questions about the future of computer science curricula, in order to generate some early thinking on these problems before they arrive.

Four Deficiencies in Computer Science Curricula

Consider the implications of the following statement: it is estimated that more than $20 billion is spent each year on software in the United States. Furthermore, the estimates are that, during the period 1975-1985, the growth in the demand for software will exceed the growth in the software labor supply and in increased software productivity by about 8%[1]. Think of it: $20 billion!

In addition to this, consider the relative shift that has taken place in the cost of computing during the past 20 years from hardware to software. Estimates are that by 1985, 85% of the cost of all computing will be in software[1].

What do these figures suggest for future trends in computer science curricula?

A brief examination of the model curriculum proposed by the Education Committee of the IEEE Computer Society[2] reveals that approximately 60% of their new curriculum is still dedicated to hardware. Although this percentage is not particularly surprising, the suggestion to me is clear: either the IEEE have badly missed the mark, or they are content to produce students who are well trained to meet 15% of the computer industry's needs.

Before the ACM or the arts-and-sciences types in computer science start feeling smug in all of this, let me suggest that they too are missing the mark. Whereas the IEEE Computer Society refuses to acknowledge that software is "where it's at", the ACM continues to show an unwillingness to come down out of the clouds of theory to see clearly what is happening in the computing industry.

In the same way that one can define the English language to consist of that spoken by English-speaking people, one can define computer science as that which computer scientists do. Consider, therefore, the activities of a computer scientist in developing a piece of software. Figure 1 illustrates two points of view as to how much time should be spent in each activity; the first is recommended by Brooks, based on his experiences at IBM[3], the second is recommended by Daly, based on his experiences at GTE[4].

Activity	*Recommended Time spent*	
Planning		*12%*
Design	*33%*	*38%*
Coding	*16%*	*15%*
Testing	*50%*	*35%*
	Brooks	*Daly*

Figure 1

Suppose we estimate that 40% of all software work is spent on program testing; then perhaps something like $8 billion per year is spent on an activity that is virtually nonexistent in computer science curricula! Before leaving this subject, let me mention that I mean here the kind of program testing which requires some background knowledge in combinatorics, graph theory, probability and statistics. I will return to this point later, when I discuss mathematics training for computer scientists.

Consider next the amount of time that a programmer typically spends in activities of program maintenance. Estimates of 50-100% are not uncommon[4-8]. More importantly, GTE's estimates are that, during the entire lifetime of a given software package, as much time is spent in its maintenance as in its initial development. In addition to

this, GTE estimates that, as an activity, maintenance is more difficult than design! Although industry used to assign maintenance to the most inexperienced programmers, they are now realizing the need to assign at least 30% of that effort to senior people[4]. Thus, from a cross-sectional point of view, perhaps $10 billion per year is spent, in one way or another, on program maintenance, another activity which is virtually nonexistent in computer science curricula.

A third major deficiency that I see in computer science curricula concerns program specification, or requirements definition. Current estimates are that the cost to fix an error after a program is tested (i.e., during maintenance) is 20 to 100 times greater than the cost to fix an error in the specification or design stages[5]. Considering the staggering amount of time that industry spends fixing errors in programs, it is a wonder to me that they haven't long since decided to invest far more time and effort "up front", especially since an estimated 50% of all errors are due to faulty design[6]. An ounce of prevention here is worth at least two pounds of cure.

One of the major problems here lies in the fact that requirements definition is a human activity, and as such deals with intangibles. Computer scientists are far more comfortable with a stated problem for computer solution than they are in stating it in the first place. They would rather design and code a poorly or loosely stated problem than spend the time and effort necessary to completely and unambiguously specify it. One reason for this is the fact that there is currently no science or methodology in requirements definition. However, recent work by Ross et al. on SA[6] and Teichroew et al.[7] offer some hope.

A fourth deficiency in computer science curricula lies in the totality of the first three deficiencies just mentioned, and more. What I am speaking of here is project management: the methodology of managing manpower and schedules for the activities of planning, specification, design, coding, testing, and maintenance. Issues like chief programmer teams, program librarians, egoless programming, establishing milestones, structured walk-throughs, management and technical reviews, and estimating costs in memory size, manpower requirements, and manpower productivity are all extremely important in reducing the alarming growth in software costs. Again, typical computer science curricula give only lip service to this vital activity.

Recommendations for Future Computer Science Curricula

I would now like to move to a different vantage point and look at the needs of industry, as perceived by Bell Laboratories, and see what implications these needs might have for future computer science curricula. At the 1978 ACM Computer Science Conference in Detroit, an address was given by Dr. M. Irvine of the Bell System, at the Department Chairmen's luncheon. In his address, he presented a slide on which he listed "desired characteristics" of a computer scientist (it was amusing to see one hundred or more department chairmen, including myself, feverishly writing them all down). They were as follows:

1. Ability to speak and write well.

2. Solid mathematical background.

3. Firm grounding in computer science fundamentals.

4. Problem solving ability.

5. Thorough knowledge of the entire software process, from start to finish.

6. Ability to design the people part of a software system.

7. Managerial ability. (Of this last item, Dr. Irvine said he would remove it if there were serious objections, but in their experience, most programmers sooner or later are asked to direct several junior programmers on some software project.)

Now let me ask two questions. Are these characteristics reasonable? What implications do they have for future computer science curricula?

It seems to me that most computer scientists would say 'yes' to the first question; they are indeed reasonable. Therefore we should ask: how could we design a computer science curriculum that would produce students having these characteristics? Permit me to give you my recommendations on each of Items 1-7 above:

1. All computer science majors, graduate and undergraduate, should be required to make an oral presentation in at least one course, either on a programming project they have done, or on some topic on which they have read a number of articles. In addition, undergraduates can be strongly advised to take a course in public speaking. With regard to writing, I would suggest that more emphasis be placed on term papers and thoroughly documented programs (external documentation in particular), and less emphasis placed on traditional in-class examinations.

2. To my way of thinking, a solid mathematical background does not consist of one year of mathematics (e.g., one term of calculus and one term of discrete mathematics). A minimum of two years of mathematics should be required. Figure 2 illustrates a recommended two-year mathematics program for computer science majors in a university having a quarter system (1 term = 10 weeks). Appropriate modifications can be made for universities having a two-semester system.

	Fall	Winter	Spring
Year 1	either → Calculus I or →	Calculus II	Numerical Methods
		Linear Algebra	Linear Programming
Year 2	Logic Prop. calc. First-order pred. calc. Theorem proving	Sets Relations Graph theory Combinatorics	Probability Statistics

Figure 2. A two-year mathematics program

3. As I see it, a firm grounding in computer science fundamentals should include:

   (i) a solid knowledge of three programming languages:

      a. assembly language

      b. PASCAL/Algol/Structured FORTRAN (choose one)

      c. COBOL/PL/1 (choose one)

   (ii) a course in machine organization/digital design, and

   (iii) a course in problem solving

To this end, I would recommend a program like the one indicated in Figure 3. I will have more to say about the courses in Figure 3 in what follows:

	Fall	Winter	Spring
Year 1	Problem solving	"PASCAL"	Machine organization
Year 2	Assembly language programming	Data structures	Analysis of algorithms
Year 3	"COBOL"	Software methodology I	Software methodology II

Figure 3. A program for Computer Science fundamentals

4. In my opinion, a course in problem solving is extremely important, and should come <u>before</u> a student learns any details of a programming language or a computer. To quote David Gries[8]:

> ". . . it is fair to say that almost none of the elementary programming books say anything about problem solving, about orderly thinking, about expressing algorithms clearly and simply. The only conclusion to draw is that the students are not being taught how to program; they are only being taught a programming language."

I admit that at present it might not be easy to teach a course in problem solving, especially considering the lack of text books on the subject[9-10]. All I can say is that I would teach such a course by working through many examples, in a top-down structured way, stating solutions in a simple, structured, algorithmic statement language. I might also mention the elementary aspects of such techniques as backtracking, working backwards, hill climbing, greedy algorithms, divide and conquer, etc.[11]

5-7. I would implement the ideas in Items 5-7 above by introducing a two-term course called "Software Methodology". This would help to overcome the four deficiencies in computer science curricula which I mentioned earlier.

In Software Methodology I, I would concentrate on such topics software management, systems analysis, requirements definition and program specification, program design methodologies, program correctness, and documentation. A major element of this course would be the development, in teams, of a large programming project.

In Software Methodology II, I would continue the development of the software process by discussing such topics as coding, program libraries, testing (of segments, modules, subprograms, and systems), complexity analyses, statistical analyses, program profiles, program maintenance, and management of maintenance. A major element of this course would be the detailed testing of projects begun in the first course, followed by extensive modifications of these projects.

<u>Issues on the Horizon</u>

Before closing I would like to make a few more jabs into your consciousness concerning computer science curricula in the future.

1. Increasing numbers of students entering college have already done some computing in high school. Soon computing will appear at the junior high and elementary levels (at least to the extent of using calculators). Looking down the road ten years, it seems that this is bound to have some effect on computer science curricula.

2. The proliferation of computer courses in other disciplines is starting to happen. I have heard of courses in computing for historians, librarians, social scientists, physicists, chemists, and biologists. Would a typical computer science faculty member want to teach such specialized courses? What effect will this have on computer science curricula? I would suggest that one implication of this will be a continued shrinking in computer science enrollments.

3. Parallel processing is here to stay, although it is much too early to say what parallel architectures will prevail or what parallel programming languages will evolve. However, I will predict that the advent of parallelism will have repercussions throughout all of computer science. This is a topic on which much could be said, but space and time do not permit it here.

4. Lurking like a growing giant on the horizon is something called "information science", something which cannot be hidden under the blanket of another course in data base management systems. The fact that a new program has emerged at the National Science Foundation in Information Science Research[12] clearly heralds the arrival of a cousin discipline to be seriously reckoned with. This new discipline has among its research topics:

   - theoretical analyses of information; its creation, use, value, transmittal or communication, storage and retrieval;
   - studies of cognitive processes underlying human information acquisition;
   - studies of language and human information processing, including query languages and information retrieval systems, and behavioral and social factors that influence the flow and use of information.

   The suggestion is strong that a large number of computer science departments and their curricula will change to encompass a broader domain called "computer and information science".

5. Finally, I must mention one more factor that threatens to have a major impact on future computer science curricula: the advent of microcomputers and personal computing. I can still recall the chill that ran down my spine when I first heard it suggested that the day will soon come when computers will be as common as cars and TVs, and will be second nature to virtually everyone. When that day comes, we computer scientists will be out of a job, at least as it is now constituted. After all, what we teach is, for the most part, pretty simple; and what we teach is slowly (but surely) slipping through the university sieve down to lower and lower levels of education. What will remain of computer science that is too big, too complex, to pass through this sieve? My guess is that much of what we now teach will disappear from computer science curricula, and be replaced by something

else: an expanded science in which computing plays a smaller but nonetheless vital role, the new discipline called "information science".

Acknowledgements

I would like to express my gratitude to Sandra Mitchell and Doris Lidtke for their helpful comments and opinions on this subject. We could talk about this for hours, and did. I would like to acknowledge the substantial influence that Gerry Weeg has had on my thinking about computer science. He was a man who had an exceptional vision of what was happening and what should happen in computer science. He also had the imagination, the persistence, and the intelligence to bring his vision to reality.

Bibliography

1. B.W. Boehm, "Software Engineering", IEEE Transactions on Computers, Vol. C-25 (1976), 1226-1241.

2. "A Curriculum in Computer Science and Engineering", Committee Report, prepared by the Education Committee (Model Curriculum Subcommittee) of the IEEE Computer Society, November 1976.

3. F.P. Brooks, Jr., "The Mythical Man-Month", Datamation (December 1974), 45-52.

4. E.B. Daly, "Management of Software Development", IEEE Transactions on Software Engineering, Vol. SE-3 (May 1977), 229-242.

5. C.C. Liu, "A Look at Software Maintenance", Datamation (November 1976), 51-55.

6. D.T. Ross, "Structured Analysis (SA): A Language for Communicating Ideas", IEEE Transactions on Software Engineering, Vol. SE-3 (January 1977), 16-34.

7. A. Teichroew and E.A. Hershey, III, "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems", IEEE Transactions on Software Engineering, Vol. SE-3 (January 1977), 41-48.

8. D. Gries, "On Structured Programming -- A Reply to Smoliar", Communications of the ACM, 17 (1974), 655-657.

9. G. Polya, How to Solve It, (Doubleday, Garden City, NY, 1957).

10. W.A. Wickelgren, How to Solve Problems: Elements of a Theory of Problems and Problem Solving, (Freeman, San Francisco, 1974).

11. S.E. Goodman and S.T. Hedetniemi, Introduction to the Design and Analysis of Algorithms, (McGraw-Hill, New York, 1977).

12. Program Announcement, Information Science Research, National Science Foundation, Washington, D. C. 20550.

PERSONAL COMPUTING AND ITS IMPACT ON EDUCATION

by

Seymour Papert
Computer Science Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02138
(617) 253-6218

[Note: This paper was taken from a tape recording of Dr. Papert's talk; it was transcribed by Roxanne Berridge, paraphrased by Ted Sjoerdsma, and edited by Anne Adair.]

I would like to begin by describing a fantasy that I was making a few months ago with the director of the Aspen Ski School.

If you know anything about skiing, you know that sometimes when the snow is just right, it is very easy to come down the slope. You can do the turns, with the snow itself practically ushering you through the right motions. Everything is so right that you think you are the greatest skier that ever walked. Later, of course, you learn otherwise.

My fantasy is based on that idea. Could you engineer a mountain with a dynamically variable slope determined by the expertise of the skier? A beginner on top of the mountain would start a very gentle slope. Off he goes. As he acquires enough confidence to stay up on his skis the charactistics of the slope change. This continues until he gets to the bottom of the mountain and he is a skier. So he learns because his environment is designed in such a way that it changes with his needs.

I don't know whether you could make such a mountain or not. The ski director and I talked a long time about what is involved in skiing

and human beings, and we couldn't decide. I doubt that you could make a mountain so that one time down would produce a skier. But I do know that the amount of time that it takes to learn to be an expert skier has been divided by 20 in the last 20 years. The level of skiing ability which everyone expects to reach (and most do) in one season of skiing is a level much higher than most people in the olden days ever expected to reach even if they went skiing every vacation, every winter, for all their lives.

This is all quite interesting, but how does it relate to education? And more particularly, to computers in education? Is skiing a good model to study with implications for education? Maybe it is time for those who are trying to reform the educational process to start looking at the success stories like the improvement in skiing education. Where the learning is much better than it used to be, one should go see what happened and attempt to transcribe that to the formal education of children.

One of the themes I would like you to keep in mind during this talk is: if you are interested in how learning happens, you should start by looking at success stories. It is amazing that if you look at the articles in the journals of education and educational psychology, you'll find that they are all about failures. The whole subject is based on how badly people learn in the environment (schools) in which they <u>should</u> learn very well. We should rather think about success stories of good learning.

I would also like us to catch the vision of aspiring to something bigger and better, of some kind of change or improvement which would make things a little better in our schools. It doesn't turn me on to think that we teach computing in schools because it is a basic skill and something that everyone ought to know. What does turn me on is the hope that the computer in some sense might be the magic mountain. What I am really interested in is not teaching about computers (however important that might be), but rather determining whether knowing about computers can really deeply change everything else. So let me tell you a story or two -- little anecdotes about learning with computers that did deeply change everything else for some people.

Before the stories I should supply the background. For the past ten years we have been involved in a project at MIT which is designed to try to bring kids and computers together. Philosophically the background is this: In 1967 I first got involved with kids and computers. Watching one of the first projects -- high school kids trying to learn to program using a Model 33 terminal and a terrible language -- it was obvious that it was great for some kids. But it was even more obvious that it wasn't the way to go. So we decided to sit back in our ivory towers and think about what could be done with the computers that <u>might</u> exist ten years ahead. (Ten years ahead is about now.)

We did decide a few things. Computers should be personal computers. Computers should be action-oriented (draw pictures, make noises, make objects move around). And there should be a programming language or a programming system that didn't just happen to be lying around, but was made for the purpose.

We set out to make a collection of all the nice features of all the known programming languages, so that we could put together a language that would allow a kid to really communicate with a computer. What we were trying to do was to find out what a first grader (or a kid of any age) could do with a computer. Out of our many years of experience, one fact became very clear -- kids of all ages can learn to program.

In case you are amazed at that statement, consider programming as talking to a computer in its language. Kids have no trouble learning to talk to people. Even two-year-olds are already talking to people with more proficiency than the kids in a high school BASIC class are learning to talk to the computer, or the kids in a high school French class are learning to speak French. Surely, there must be a way to get the kids to learn to talk to computers the way they learn to talk to people. Why should it be harder to talk "computer" (as kids say) than it is to talk English? (This becomes a guiding theme.) We decided that one reason why it seems harder is that there are no good things to talk about. So we put graphics on the computer, which certainly is something to talk about. Kids can now make pictures and do things with computer pictures that they can't do with pencil and paper pictures. They can get precisions, get repeatability, depict motion, change color, and do all sorts of other things that can't be done with pencil and paper.

We suddenly came into quite a number of dimensions of the educational enterprise, or what it is to learn. For example, there is something that is called social resonance. It became very clear that just because kids spent a lot of time watching television in the home, making a picture on the television screen gives them a special kind of charge. A kid who is taking over the television screen so that he can make pictures himself is moving into the space age. And something of that drive makes all of us want to buy little calculators (that we could perfectly well do without) because we would each like to have, as our very own, a piece of that space age technology. This carries over to our kids, again a sort of social resonance. The kid feels like he is doing something meaningful, like those guys he sees in the space center controlling a moon rocket. The computer, just because it is high technology, provides a certain bridge between the learner and the world.

I would like to have you examine an even more important bridge, the one between the learner and himself, between what is being learned and what is deep down inside him. I will try to get you to sense what I mean, even though I cannot fully explain it.

We were all sort of led bit-by-bit into a view of education that is more Freudian than Piagetian. Jean Piaget is a psychologist and an epistemologist. He thinks of mathematics in a very logical way, rigorous and precise. And I think that he is right in a lot of his insights. What I am trying to say is that there is a model of mathematics which doesn't have much to do with my feelings. The general view of mathematics in our culture is that it is something separate from our bodies, cold and discorporate.

Let me show you a different view of mathematics and an example of what it is like. Let's go back to those children learning to make pictures on the television screen. How do you go about making a

picture on the television screen? The child is creating it, controlling it, intellectually interacting with it; he is not just pushing buttons and flashing things on the screen as if by magic. Thus we became involved in the enterprise of how to think of geometry. After a lot of false starts, we came up with what I think is the absolutely right way to think about geometry. This is the concept of "turtle". "Turtle" is a differential equation -- differential geometry in computational terms.

Let me explain how we present it to a child, and then maybe you won't worry about it being differential geometry. I believe that what we are talking about has a certain great importance, not because it is capturing an idea that we invented, but because it is capturing ideas of science and bringing it into a new view -- and the new view is possible only because of the computer's presence. It is this creation of new visions of old things that is one of the magics of the computer.

We bring in the child and introduce him to a little cybernetic animal called a "turtle". The turtle is controlled by the computer and the computer is controlled by the child. The turtle is positioned somewhere and is looking in some direction. This is his state. To change his state, there are two different change operations: forward and turning. Forward causes him to change his position without changing the direction that he is looking. The turning operation allows him to turn to the right or the left without changing his position. These two operations (coordinates) can change the two components of the state independently and thus the child can get the turtle to perform any maneuver. If the child equips the turtle with a pen, then sets it on paper, the turtle could draw things. (Alternately, one could introduce the child to the turtle on a display screen, which could obey the same commands, but now draws pictures with the speed of light.) Understanding the geometric concept of the state of the turtle, with its two kinds of change operations, gives one a kind of mathematical knowledge which provides power to do things.

However, I don't want to emphasize that aspect, but rather a kind of learning that I call "body syntonic learning." Let me introduce this with an example. Imagine a child who has learned to drive the turtle; he knows how to say forward and how many units; he knows how to turn it and how many degrees of turn. By and large the child has been drawing straight line figures. One day the child asks "how does it draw a circle?" We ask the child to figure this out for himself. We suggest that the child pretend that he is the turtle and walk in a circle and try to describe his movements in turtle language.

This is where we connect the geometry to the child. The turtle is especially suited to serve as what has been labeled a "transitional object." The child comes along knowing a lot about "body geometry." He knows how to walk around in space. He has a lot of geometric knowledge, but it's inside and he can't verbalize it. On the other hand, there is this abstract thing called mathematics which you want him to learn. And in between is the turtle -- a mathematical object, but enough like a person for him to identify with and to transfer his knowledge of himself to. Because the turtle moves like he does, the child can apply the same description and sort of talk to it through the computer.

So we have seen two ways in which the learning of mathematics is different from the traditional. One way is social resonance. The other way is resonance with deep structures inside the person. In fact, this is where Freud and Piaget come together.

Let's return to the circle. When the child walks in a circle, he goes forward a bit and right a bit, forward a bit and right a bit, repeating this until he returns to his starting point. So, too, when he tells the turtle what to do: forward one, right one, repeated as often as necessary until it results in a circle, at least from a point of view that defines a circle as a curve of constant curvature. (You see that the curvature of a curve is how much you turn to how much you go forward.) If the child goes forward ten and right one, it would be a much flatter curve, and would produce a much larger circle that would look like a polygon. He could make a spiral by gradually decreasing the curvature. It is a very impressive thing to make a spiral and then to get all sorts of crazy pictures by putting spirals on top of one another. Mastering this idea of curvature by thinking of curves in terms of generating little changes is an extremely powerful idea. It is perfectly graphicable by these small children, and also very usable. This last aspect is the third way that this mathematics is different from school mathematics -- its social resonance, its resonance with one's own body, and its usefulness. It can be used right now to do something exciting.

Here, you might say, is applied mathematics instead of pure mathematics. It is a curious thing that the way we teach mathematics in the elementary school is very very pure. Historically, that is a very funny thing. Historically, mankind first made mathematics a part of doing other things, like navigating, commerce, magic, religion, and all sorts of architecture. Only bit-by-bit did that beautiful thing called pure mathematics become abstracted out. I think this is a right model. And it is the model that Piaget teaches us, too, if we look carefully. The right way to learn mathematics is first learning it as a concrete applied thing that actually works in the world, and then gradually we get to pure mathematics.

So here is a model for a computer making a radical difference to the world. It is not just able to change how we teach the same algebra and the same equation in the same number of years. It can change the entire relationship to mathematics. And not only mathematics. I chose mathematics only because, in our society, mathematics is so split off and dissociated from everything else. But the computer could have the same impact on learning French. And what we see in the French classroom or in the mathematics classroom tells us nothing at all about difficulties we have in learning these subjects.

We have seen so clearly in our laboratory what happens in ideal conditions. What we have seen in so many instances in our children is that those kids have taken to mathematics so easily, have learned with such facility, that they are being slowed down instead of accelerated by the whole school process. The teachers know that too, but they don't know what to do about it.

I would like to give it a little theory and talk about dissociate learning. For most people in this culture, mathematics has no relation to anything they enjoy, or love, or do, or appreciate. It is

in a box apart. We are told that some people (like scientists) use it in some magical way to make moonships of which you know nothing. This is dissociate learning.

By syntonic learning, I mean "it goes together with". Children learn a tremendous amount of stuff by syntonic learning, including a tremendous amount of mathematics. They don't only learn it, they invent it. Piaget is important for investigating the thinking of small children and finding some remarkable discoveries. Piaget has filled books and books and books full of things that children have discovered. If it were totalled and made into a curriculum, teachers would pass out in horror at the thought of the immense task. But the same would be true if you considered what a child learns in order to speak his mother tongue, or to manipulate his parents, or to know what is a funny joke. The fact is that children, until they get to school, are avid, eager, successful learners. Then, when they come to school, they learn that learning is not so good.

The point is that children do acquire, learn, construct, reinvent (or whatever the word for it is) a vast amount of mathematical knowledge (and other kinds of knowledge) before they ever come near the classroom. And this is not knowledge that is acquired in a dissociated way. So I would suggest that syntonic learning may be another good name for Piagetian learning, which is a kind of learning that happens without being taught, without a curriculum, quiz, or grade. And there is certainly a fantastic amount that is learned this way.

Why not everything? Why are there some things that seem to require such a different learning process. I would say that one of the reasons is that, although we have always known in principle what to do about it, we haven't had the technological infrastructure to realize the things that we've already done. Now the computer has brought us the technological infrastructure that can make possible a real intervention in the learning environment, one in which syntonic learning can be generalized to cover a much greater area. I hope we can make the most of it.

HUMAN FACTORS:
IMPACT ON INTERACTIVE COMPUTING

by

Walter J. Doherty
T.J. Watson Research Laboratory
International Business Machines
Box 218
Yorktown Heights, New York 10598

This presentation will describe the human side of interactive computing as seen in the computing center at IBM's T.J. Watson Research Laboratory in Yorktown Heights, New York This computing center has been a SHARE member (PK) since 1958.

The mission of the Watson Research Lab is to provide advanced computer services; to encourage innovative uses of computers; to make computers easy to use; to educate users as to what is available and how it can be used; to undertake selected development projects; and to measure, analyze, and evaluate computer services in order to improve and extend them. We find that complexity is the single biggest problem facing the end user today. Since 1968, we have looped on the last two items in our mission. We made the decision at that time to stand at the man-machine interface and look outward to study the impact on people of the computing function which has evolved over time. We learned of a number of technical and administrative barriers which got in people's way. We worked hard to remove those barriers.

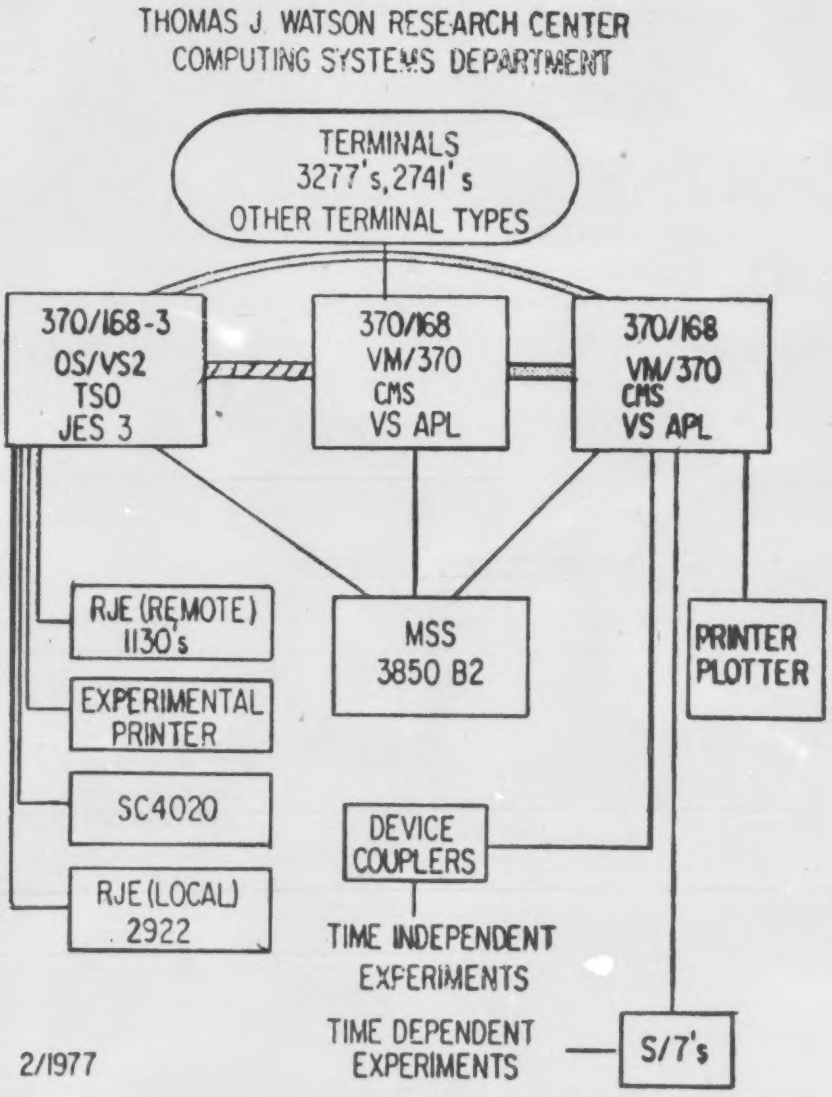
Upon removal of these barriers, we can make the following observations.

The value of today's computing systems to their end users is in their data, programs, and procedures. Libraries of thousands of programs exist in all major installations. The user interface is simplified by combining programs together and burying default parameter specifications inside procedures. Procedures outnumber all programs written in conventional programming languages.

The issue of integrating the data resource into the enterprise (building bridges to tie the different subsystems together) will probably be the dominant driving force in computing over the next ten years.

Figure 1 shows the current picture of the computing configurations at Yorktown.

Figure 1



Notice that all systems can communicate with each other and use the MSS as a common back end. Terminal growth has been rapid, especially since the advent of the IBM 3270. Laboratory automation, program development, and text processing have all grown rapidly as well. VM/CMS is the interactive interface which is the most popular in our laboratory.

At Yorktown, the computing systems have moved from being specific tools to solve specific problems to being a general resource which extends people's memory and reasoning power across the entire spectrum of tasks they do. This is a new dimension of computing use. It requires the removal of the barriers we encountered to achieve the levels of productivity we have today.

Figure 2 shows the growth in end user time relative to computing costs from 1973 through 1977 at Yorktown. The uppermost line shows the growth by a factor of 8 in monthly ' connect' hours. This includes dormant end user time, active end user time, and special function virtual machine time. The next two lines down show the growth in end user time at the terminal. The uppermost line shows a growth from 8,000 to 64,000 hours per month. It is a reasonable upper limit on the amount of time that the interactive systems were influencing our people in their work. The lower line, showing a growth from 5,000 to 38,000 hours per month, is an estimated lower bound on the amount of time that our interactive systems had a direct influence on people's work. The next two lines down show the total computing cost relative to the end user cost, and the hardware cost relative to the end user cost.

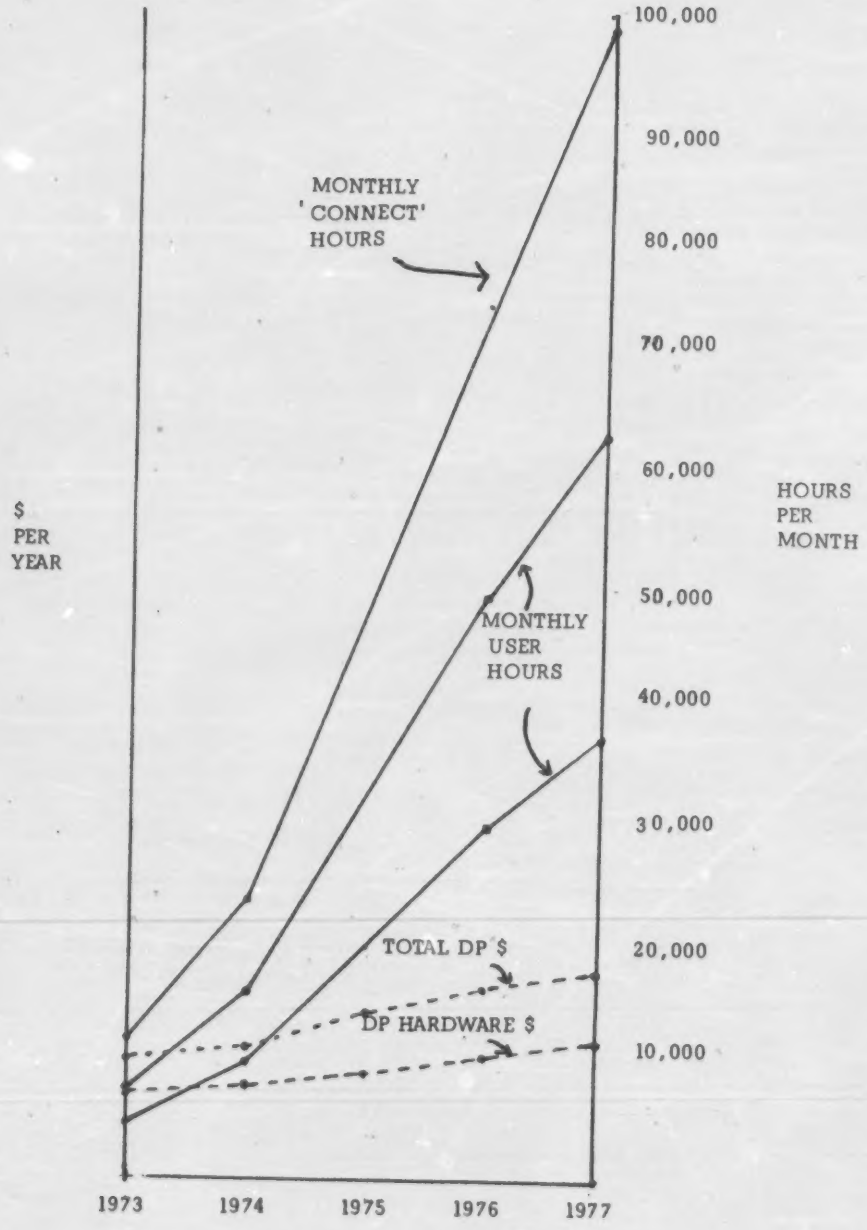
Thus, small errors in DP equipment planning can result in sharp losses of end user time due to overload.

Figure 3 shows how complete the growth of interactive computing has been in penetrating the different departments we have in Yorktown. It shows the number of people in each department who spent many hours of their time working interactively in November 1973, and again in November 1976. For example, the first column shows that department 41X, our administration department, had three people in 1973 who spent from 20 to 40 hours per month of their time working interactively. It also shows that there were 38 people in that department who spent from 20 to 40 hours per month of their time working interactively in November 1976.

Yorktown processed at least 7,000,000 and at most 11,500,000 interactions per month in November 1976. By November 1977 that range had grown to at least 9,000,000 and at most 15,000,000 interactions per month.

Figure 4 shows the number of people in the Laboratory who spent different amounts of their time working interactively in the month of November 1976. Today, over 25% of our Laboratory is working interactively at each instant throughout the day. Over 75% of our Laboratory works interactively in each week.

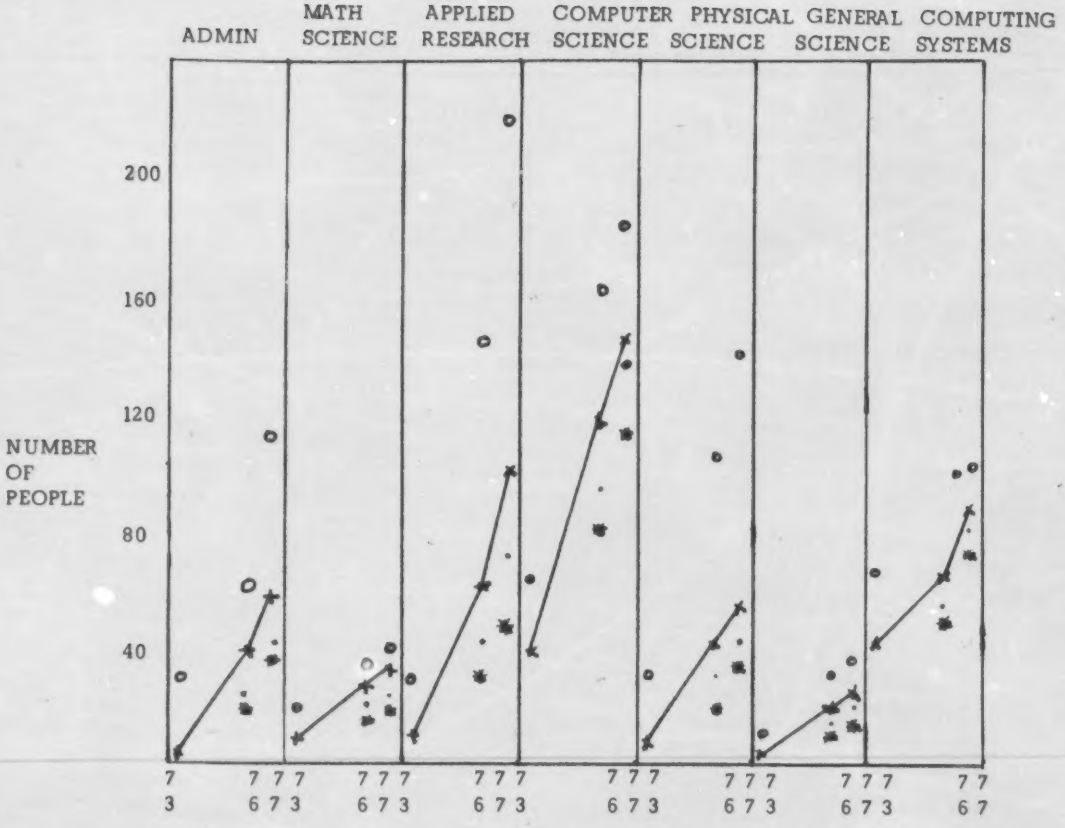
Figure 2



IBM T.J. WATSON COMPUTING CENTER
COMPUTING COSTS vs. USER COSTS

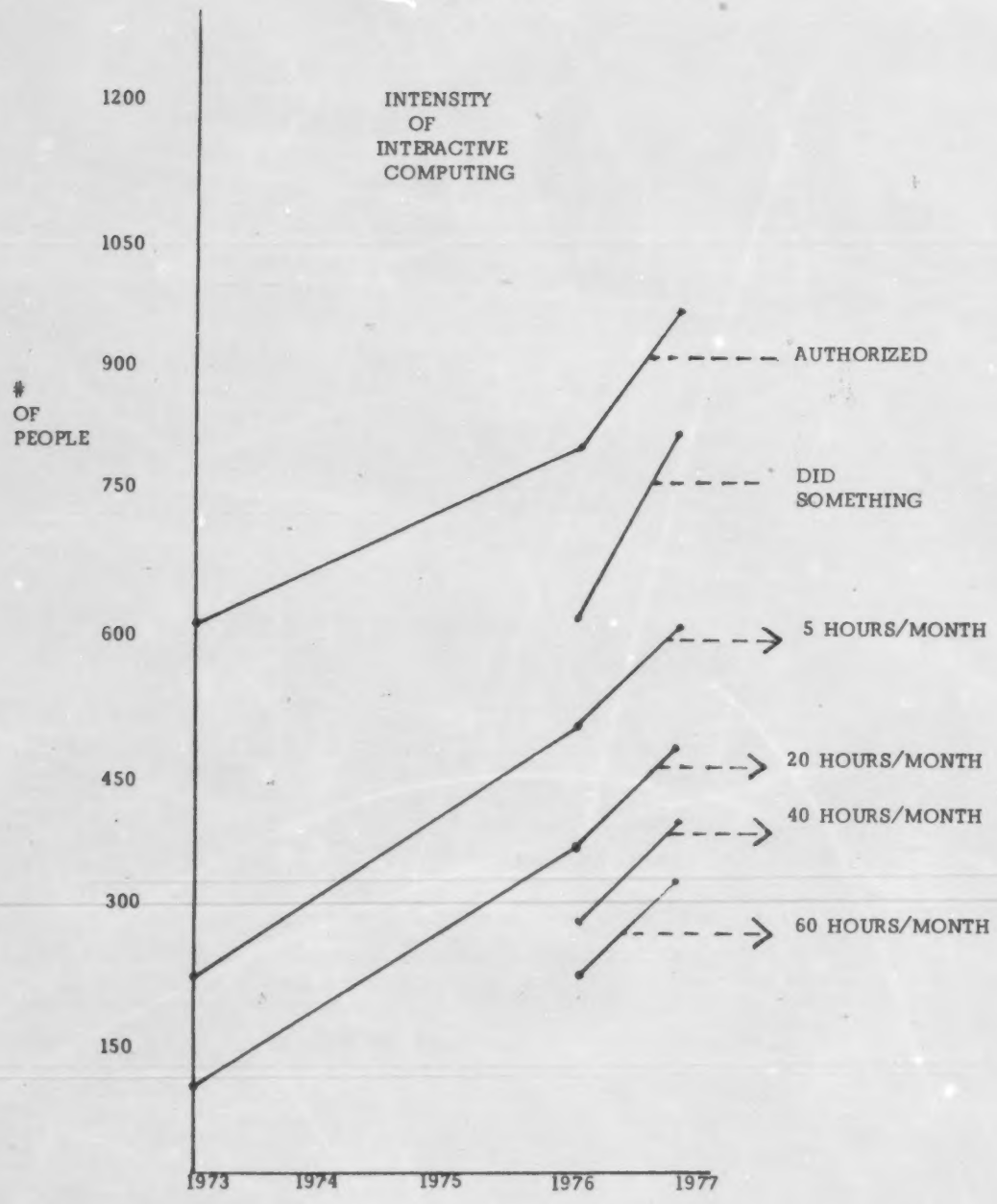
page 51

Figure 3



MONTHLY GROWTH OF INTERACTIVE COMPUTING
PER DEPARTMENT
1973 - 1977

Figure 4



YORKTOWN

page 53

Figure 5 shows the difference in intensity between an interactive and a batch environment. Because interactive computing is so much more intimate in its impact on the end user, we have learned that functional availability must be targeted for 24 hours per day and 7 days per week. If information is not available from a computing system when it is needed, then people do not keep all their data online. This also implies that maintenance must be time shared.

Figure 5

INTERACTIVE	BATCH
CONTINUOUS	DISCRETE
5 HOURS / DAY	1/4 HOUR / DAY
EVOLUTIONARY	STATIC
200 COMMANDS / DAY	9 STEPS / DAY
LOAD RANGE 100,000 : 1	LOAD RANGE 100,000 : 1

In 1972, we saw that the number of commands per person doubled in just five months, when counted after procedure expansion.

At the SLAC (Stanford Linear Accelerator Center) installation today, using the WYLBUR interface, there are an average of twenty-five commands executed for each command typed at the terminal.

People learn how to use a computer to perform some function for them. They then ask the computer to remember how to do that. As people grow in experience with such a system, they combine functions to raise the information processing bandwidth between themselves and the computer. It is hard to forecast how long this growth will continue. It seems evident that the quality of the command procedure facility, which is the software glue to help this growth, is critical to user growth.

Figure 6 shows the number of different file types on one of the VM/CMS systems at Yorktown. Of the 800 people who use this system, this chart shows the files belonging to 164 of them. In some sense, that's about 20% of the users of this system at Yorktown. Notice that 162 of the 164 users had a total of 6074 EXEC files. If we look at all other forms of programming language, we get a total of 3000 programs in source form. EXECs are used to raise the level of the man-machine interface by combining programs together. They reduce complexity by allowing many parameters' values to be decided automatically. They reduce errors by eliminating the need for people to reenter the same commands and parameters time and again.

As people learn how to do more and more things with computers, EXECs help that growth. EXECs are probably the single most important facility in the system to aid user growth.

Editors are the single system component where the computer user spends the most time. Estimates are that 50 to 80% of people's time is spent editing text data or programs. 75% of all commands typed are for editing.

Figure 6

1	EXEC	6074	23.62428	102	99.78049
7	SCRIPT	2397	9.322132	115	72.56097
8	TEXT	1758	6.837008	114	69.51219
10	MODULE	728	2.831252	99	60.36584
6	ASSEMBLE	656	2.551238	73	44.51219
20	MAIL	261	1.01505	68	41.46341
45	PLI	1496	5.8256e7	64	39.02438
16	DEFAULTS	72	0.280014	58	35.36584
127	FORTRAN	723	2.811807	56	34.14633
19	$DIRECT	68	0.2644576	49	29.87804
52	MACLIB	68	0.3206683	44	26.82925
22	VMAPLWS	350	1.361178	42	25.60974
95	MEMO	198	0.7700385	41	25.
15	ARCHIVES	56	0.2177886	40	24.39024
21	SYNONYM	40	0.1555633	40	24.39024
14	CONSOLE	52	0.2022323	38	23.17073
104	SYSIN	446	1.73453	36	21.95122
193	TERMTEXT	99	0.3850192	30	18.29268
53	TXTLIB	46	0.1788676	30	18.29268
17	VSAPLWS	147	0.5716952	30	18.29268
148	JCL	253	0.983938	28	17.07317
277	PLIOPT	146	0.5676061	28	17.07317
23	OLDMAIL	27	0.1050052	25	15.2439
84	PLS2	272	1.05763	20	12.19512
226	DIAG	24	0.9333795E-01	16	9.756097
96	MACRO	177	0.6883677	12	7.317073
25	LISP	164	0.6376096	9	5.487804
451	PIC	40	0.1555633	9	5.487804
460	MVS	66	0.2644576	8	4.878048
112	OS	35	0.1361179	8	4.878048
144	PICTURE	35	0.1361179	8	4.878048
424	GRIN	56	0.2177886	6	3.658536
450	LABEL	39	0.1516742	6	3.658536
18	LISP370	76	0.2955703	6	3.658536
719	KGRAPH	85	0.3305721	4	2.439024
345	KSCRIPT	27	0.1050052	4	2.439024
499	INCLUDE	28	0.1088943	3	1.829268
110	SBS	305	1.18617	3	1.829268
94	BCPL	63	0.2450122	2	1.219512
723	CLIST	115	0.4472845	2	1.219512
743	FTO5F001	94	0.3655736	2	1.219512
24	LISPLIB	31	0.1205615	2	1.219512
1668	RUNOFF	36	0.140007	2	1.219512
210	PISLIB	51	0.1983432	1	0.6097561
1557	FONT	34	0.1322288	1	0.6097561
285		7	0.2722358E-01		0.
223	$MONDAN$	15	0.5833624E-01		0.
441	A	82	0.3189048		0.
123	ACCESS	7	0.2722358E-01		0.
119	AEL	33	0.1283397		0.
781	APAR	9	0.3500174E-01		0.
1669	AREADESC	52	0.2022323		0.
1670	AREADUMP	52	0.2022323		0.
565	ASM	43	0.1672305		0.
782	ASSEMBSV	25	0.9722704E-01		0.

Editing programs have historically not received much attention as formal computer software products, and have historically been neglected by system development groups. However, we find ourselves with a great proliferation of editors today. Carol Thompson's User's Guide To The Research Context Editor--Redit (IBM Research Report No. RA 28A) contains an excellent history of the evolution of one such editor program.

It seems that the human factors associated with editor programs are therefore of great importance.

Figure 7 shows a picture of the response time to a single small program which ran every 20 seconds in the spring of 1971. Each 20 seconds, it woke up, looked around at the other users on the system, and recorded the response time that it received from the system. This is really a 3-dimensional graph. Each plot position is a number indicating the number of times that operating condition was found to exist.

Notice that as the number of simultaneous users increases, the response time steadily and sharply declines. Thus, if we simply try to maximize the number of simultaneous users, we find that we are maximizing the number of people to whom we are presenting a sharply degrading picture of service.

Remembering how rapidly people grow in their use of programs, it is clear that a given number of people will seldom ask for the same work to be done.

If we recall that user growth results in a sharply increasing number of programs invoked per person, and that service degrades beyond some load threshold, we should realize that the number of simultaneous system users is a misleading measure. It is indeed a measure of potential liability. It is not clear what else it measures.

We find that most genuine performance improvements lead to a direct reduction in the number of simultaneous system users. This is accompanied by an increase in the number of interactions processed. There is also an increase in the number of people served over a longer time such as a day or week. The simultaneous congestion has decreased.

Figure 8 shows the profound influence that system response time degradation can have on user behavior. If we break an interaction into two parts, the system response time (SRT) where the system is processing a request for the user, and a user response time (URT) where a user is inputing his or her next request to the system, we find that each second of system response degradation leads to a similar degradation added onto the user's time for the following request.

Figure 7

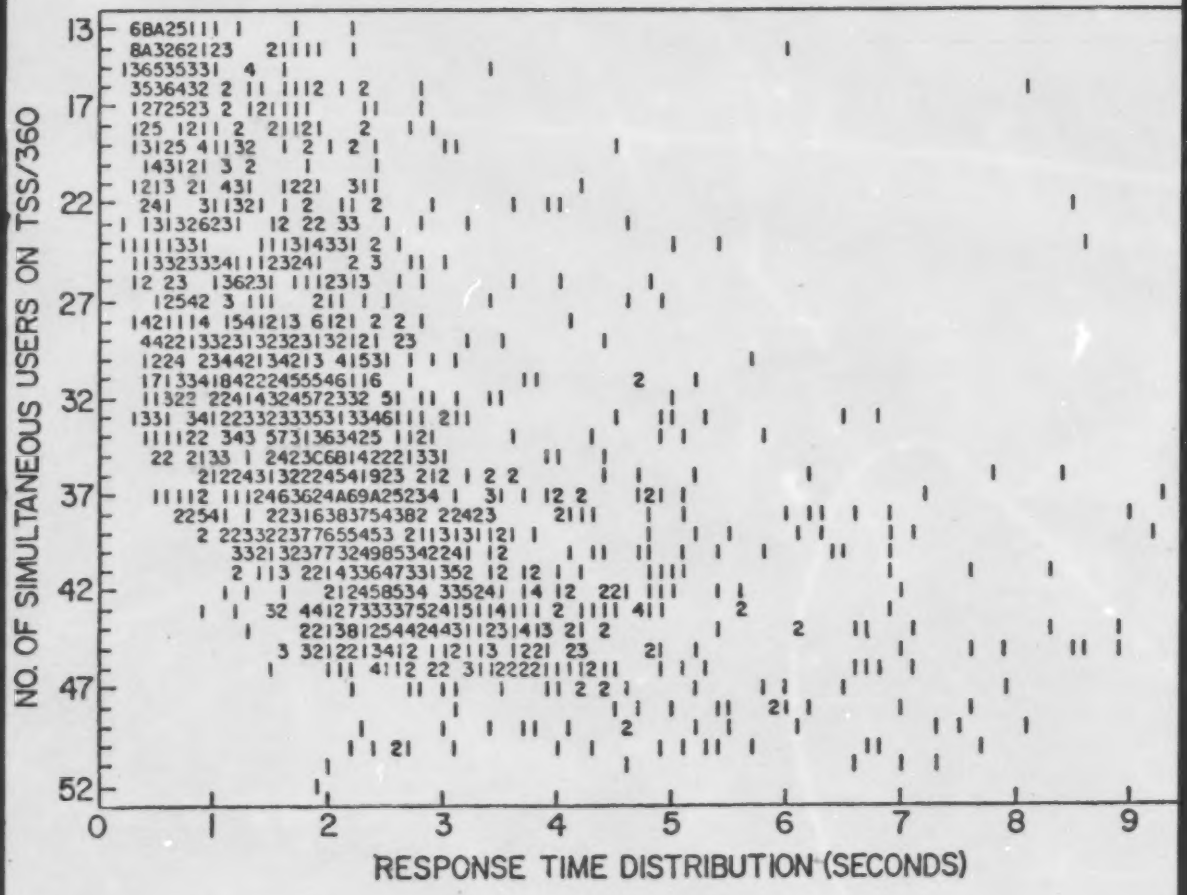
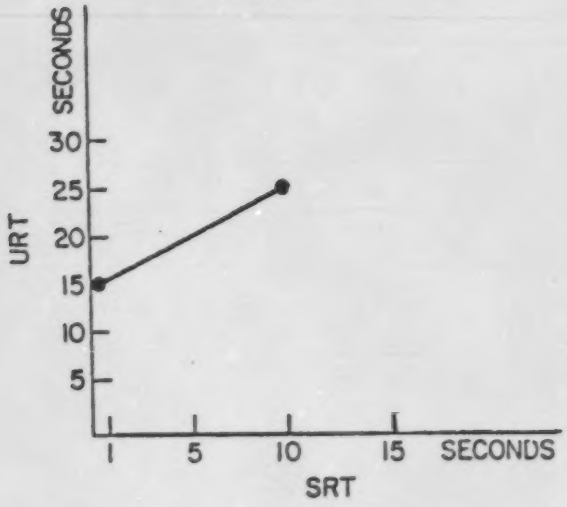

NO. OF SIMULTANEOUS USERS ON TSS/360
13
17
22
27
32
37
42
47
52
0
1
2
3
4
5
6
7
8
9
RESPONSE TIME DISTRIBUTION (SECONDS)

Figure 8



This phenomenon seems to come from people's attention span. The traditional model of a person thinking after each system response appears to be inaccurate. Instead, people seem to have a whole series of actions in mind, contained in a short term buffer in their heads. In fact, they are continually thinking. Disruptions to their thought processes are generated by delays in SRT. This results in their short term memory buffer being reset.

This phenomenon was first discovered by S.J. Boies at Yorktown in 1971 while doing normative studies of interactive computing. We had recorded every interaction on TSS for three years. It included a wide range of human tasks. The phenomenon was consistently repeatable across most types of work.

In 1936, controlled behavioral studies showed that three things happened to people in a stimulus response situation when the stimulus became both long and erratic. First, they slowed down in their work. Second, they became emotionally upset. Third, they made more mistakes.

If we were to tolerate a system response time as long as two seconds in our Laboratory today, it would cost us a minimum of 36 million seconds per month of lost people time. That is 10,000 man-hours, or 60 people lost full time for the month. It is our objective to keep 90% of our interactions to a response time of .5 seconds or less. Subsecond response time is an important human requirement.

Figure 9 shows the distribution of user response time (URT). Others have called this THINK time. The mode of this distribution is 2 seconds. The median is 8.5 seconds and the mean is 12 to 15 seconds, depending on when you terminate the data points. If all end points, such as the two hour lunch break or the time when people forgot to log off, are included, then any average is conceivable.

However, what is really being measured, then, is how long the system stayed up. By truncating after two minutes, we find an average URT of 12 to 15 seconds. This distribution has held up for twelve years now. The high frequency of points in the two second range reflect the fact that people scan far more information than they read, and read more than they write. There is a real need for computing system response times on the order of one or two tenths of a second.

Figures 10 and 11 show the load on a S/360 Model 67 from running a few programs. They are a conservative 2-dimensional picture of load. CPU time is on the horizontal axis and main storage requirements in 4096 byte pages are on the vertical axis. The shaded area is supervisor state time, and the clear area is problem state time. The area under each rectangle is a conservative measure of load. It is conservative because the I/O dimension and the paging overhead dimension is missing. The difference in area varies by five orders of magnitude from the smallest to the largest. Almost all the variation in load comes from the amount of data being processed and not from the kind of processing.

Figure 9

FREQUENCY OF USER RESPONSE TIME

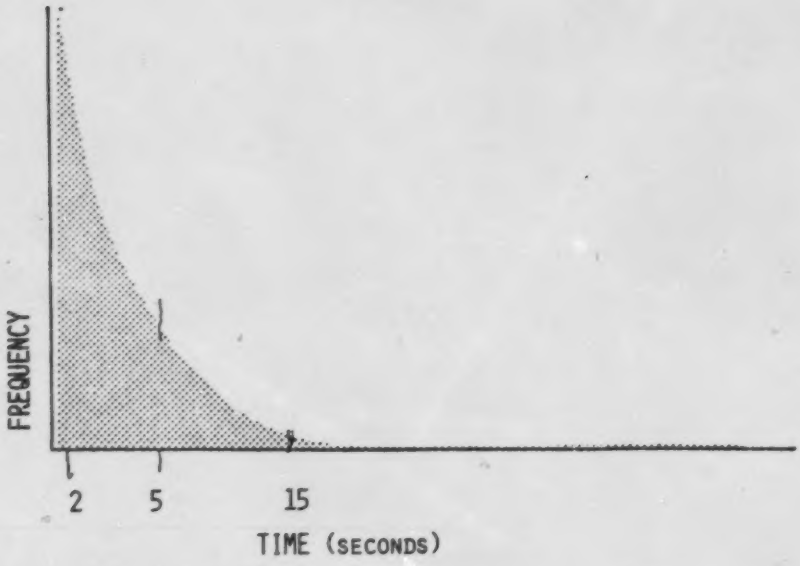


Figure 10

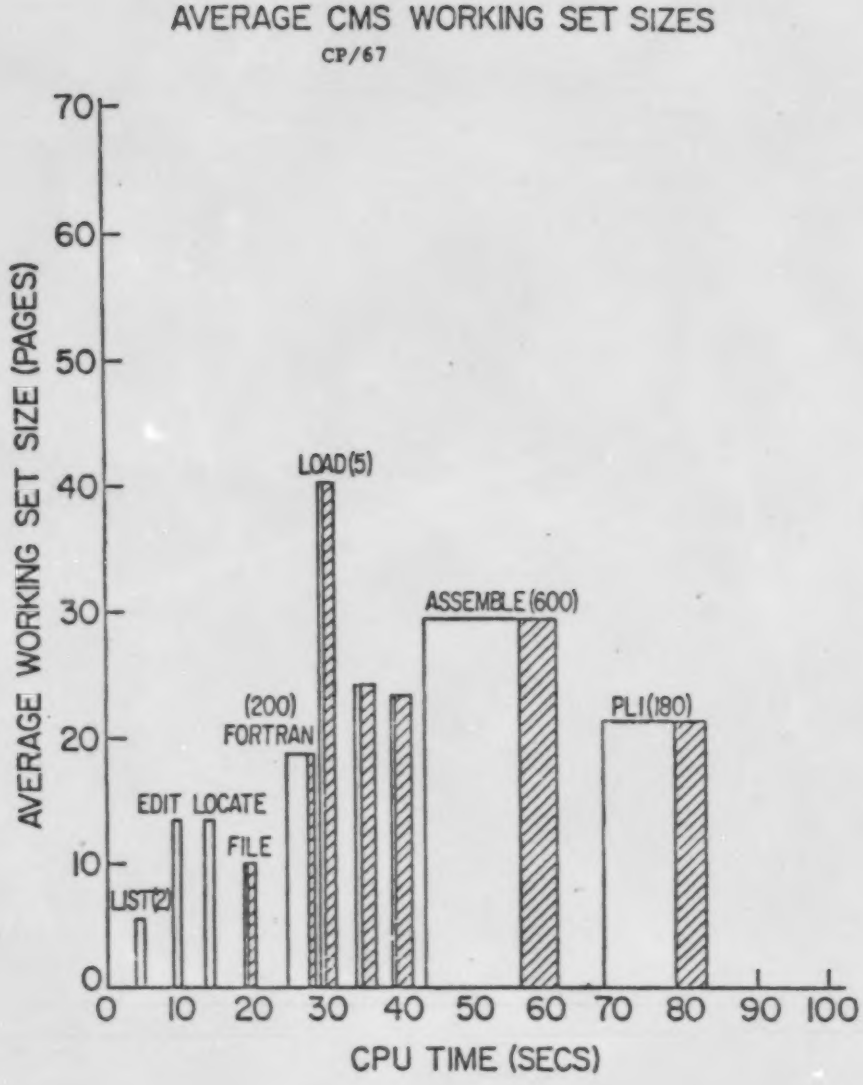
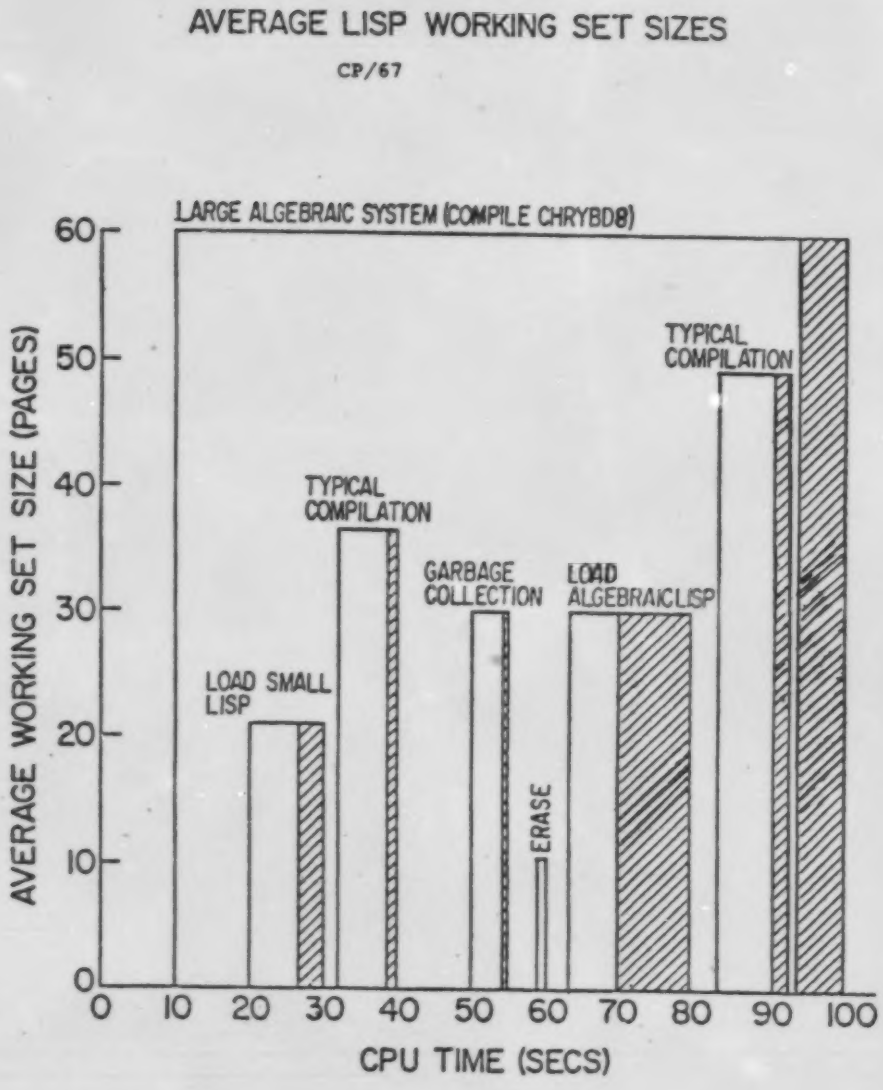

AVERAGE CMS WORKING SET SIZES
CP/67
AVERAGE WORKING SET SIZE (PAGES)
CPU TIME (SECS)
0
10
20
30
40
50
60
70
80
90
100
LIST(2)
EDIT
LOCATE
FILE
(200)
FORTRAN
LOAD(5)
ASSEMBLE(600)
PLI(180)

Figure 11


AVERAGE LISP WORKING SET SIZES
CP/67
LARGE ALGEBRAIC SYSTEM (COMPILE CHRYBD8)
TYPICAL COMPILATION
TYPICAL COMPILATION
GARBAGE COLLECTION
LOAD ALGEBRAIC LISP
LOAD SMALL LISP
ERASE
AVERAGE WORKING SET SIZE (PAGES)
0
10
20
30
40
50
60
0
10
20
30
40
50
60
70
80
90
100
CPU TIME (SECS)

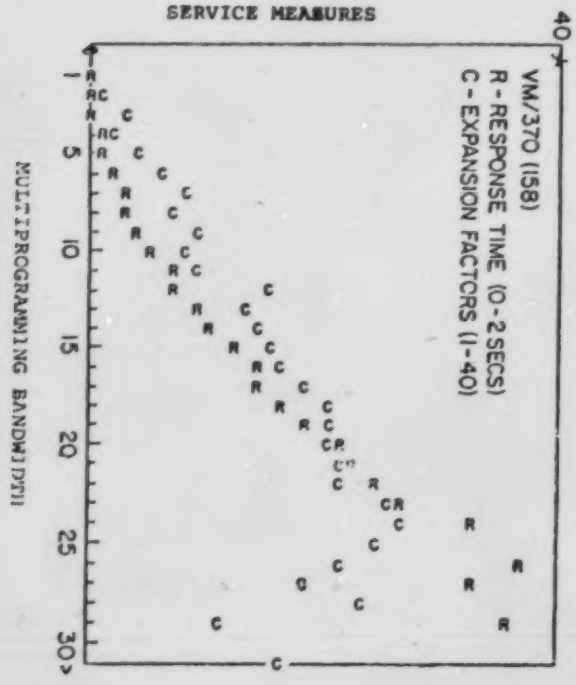
A classic problem in our industry has been the problem of benchmarking. This has arisen as a problem by the confusion between load and function measures. For example, one installation said that they had characterized their load. They had a beautiful functional distribution of their work, but had only accounted for 2% of their load.

Load (that is, demand on the CPU, I/O, and main storage) arises mainly from the amount of data being processed rather than from the nature of the processing. Variations in load on the order of five orders of magnitude are regularly seen for editing as well as compiling.

Measures of function are useful for productivity studies. They are of very little use in capacity planning.

Figure 12 shows service measures for two classes of work as seen on a S/370 Model 158 in one afternoon in July 1974 at Yorktown.

Figure 12



PTS FOR C	PTS FOR R
53	112
90	681
113	1522
152	2123
127	2569
113	2931
101	2772
121	2998
132	3425
147	4062
170	5081
206	6265
252	7729
309	9616
315	11678
347	13410
322	13809
347	14009
291	13101
265	11915
216	9836
151	7557
89	6329
68	3615
39	2211
23	1227
12	607
3	327
3	120
1	66
1	75

The horizontal axis shows the level of multiprogramming. It is the sum of the ELIGIBLE and DISPATCHABLE lists in VM. It is the parallelism in the input stream of requests. The vertical axis shows two different classes of service measures, depending on the nature of the work being done. For those interactions which could be completed within 200 ms. of 158 CPU time, the service measure is RESPONSE TIME, and it ranges from 0 to 2 seconds. For those interactions which were bigger, we use the EXPANSION FACTOR as the service measure. This is a multiplier that measures how much longer the work takes due to the sharing of the machine than it would in a stand-alone environment. The numbers across the top are the number of interactions in each class of work at each level of multiprogramming.

Notice that most of the interactions took place at a level of multiprogramming between 15 and 21. There is more simultaneity in the input stream than in the hardware.

Notice also that 97 to 98% of the interactions fell in the light load class of work. That is 97 to 98% of the people time, the most expensive commodity. The remaining two to three percent of the interactions accounted for 80 percent of the total demand on the CPU, I/O, and main storage.

By selectively detecting and scheduling the few large tasks, we can assure the remaining 98 percent of the interactions will have excellent response.

This has been done in the Resource Management PRPQ in VM/CMS.

Figures 13 and 14 show the distribution of CPU time over the user population at Dartmouth and in a pure batch environment at Yorktown. The horizontal axis shows the cumulative percent of the people using each system. The verticial axis shows the cumulative percent of CPU consumed by those users over a month. Notice that 95 percent of the users use a total of 25 percent of the resource. If we further examine the top 5 percent of the users we find that 5 percent of the programs and data they run account of 85 percent of their load. Again, a very small percent of the interactions account for the overwhelming majority of the load.

By putting a small effort into tuning these few users' large programs, we find large service gains can be achieved in short time periods. This seldom results in less overall load since you are taking a machine limited activity and making it less so. Most people simply run their programs more then. However, their service is improved sharply, and the granularity of the load they impose on others is smaller, thus giving others better service as well.

It turns out that I/O has a similar distribution while main storage is slightly flatter.

Again, scheduling can be, and is, effective in taking advantage of this load distribution.

I have seen many different systems in many different environments. All of them, except ACP, have the same load distribution. This is a reliable and repeatable load curve distribution.

Figure 13

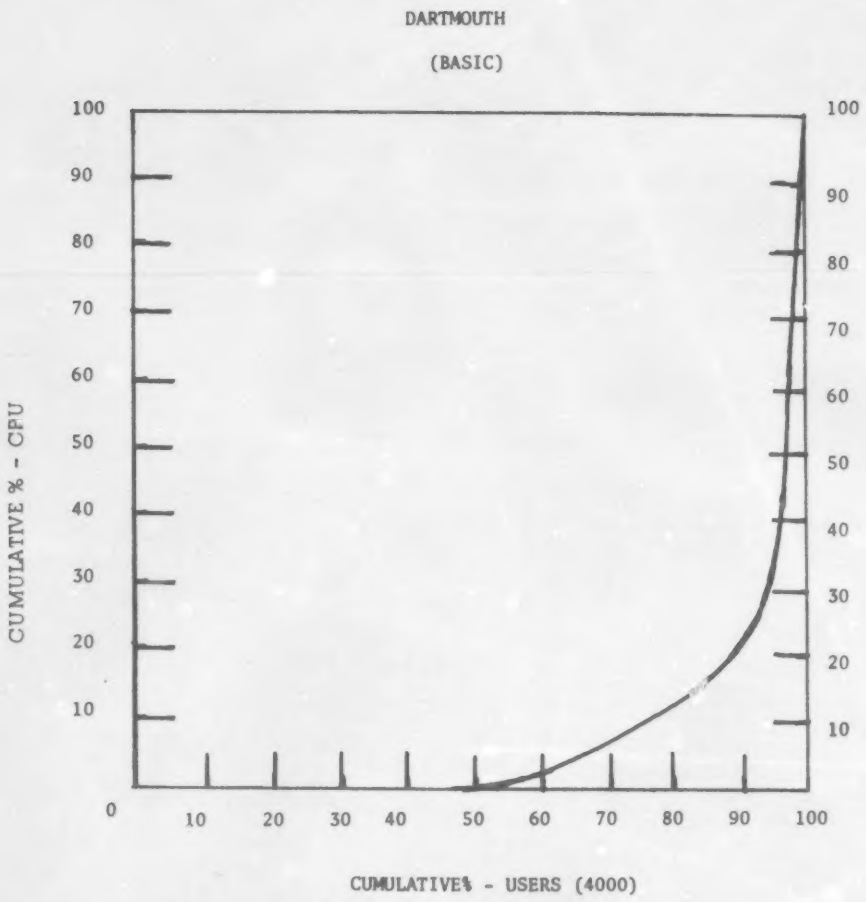

DARTMOUTH
(BASIC)
CUMULATIVE % - CPU
100
90
80
70
60
50
40
30
20
10
0
10
20
30
40
50
60
70
80
90
100
CUMULATIVE% - USERS (4000)

Figure 14

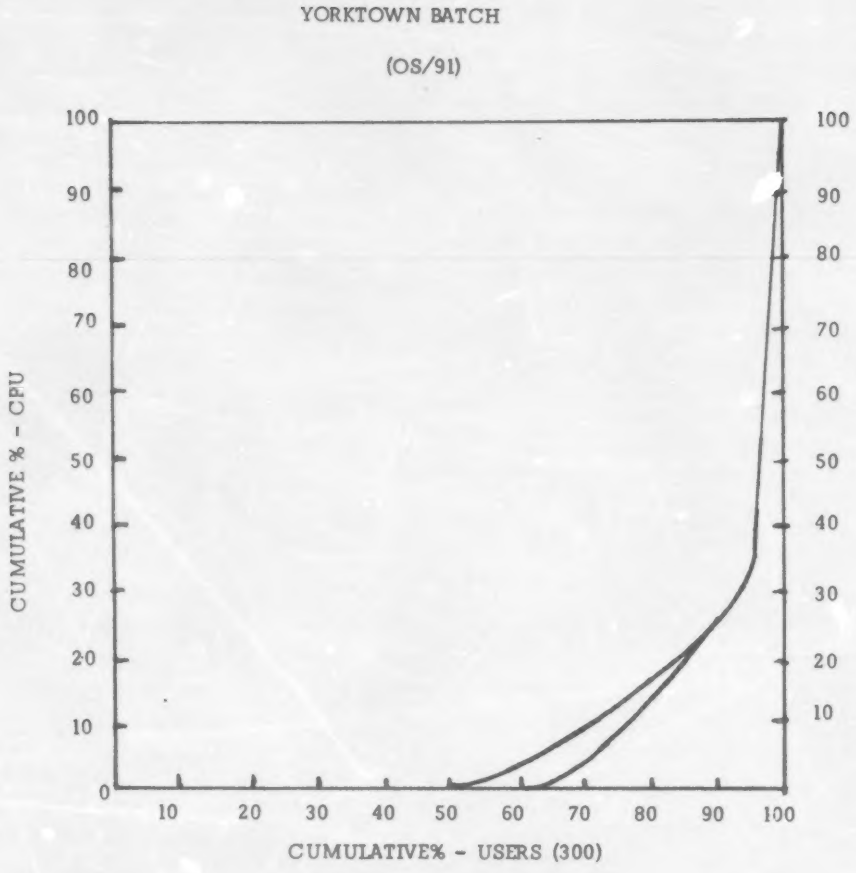

YORKTOWN BATCH
(OS/91)
CUMULATIVE % - CPU
CUMULATIVE% - USERS (300)
0
10
20
30
40
50
60
70
80
90
100

Figure 15 highlights the impact of scheduling on the man-machine interface. It shows the impact of the Resource Management PRPQ for VM/370 in a benchmark environment as it would apply to people. The columns titled WHEELER show the results of running with the PRPQ in an overloaded environment. The columns titled RELEASE 3 show the same overloaded environment without the PRPQ. Notice that 50 percent more interactions can be handled by the PRPQ in this overloaded environment. And the response time at the 90th percentile is 20 times better. Notice also that the base system handles more interactions with 60 people than with 80. By allowing those extra 20 users onto the system in that overloaded environment, we observe that we have effectively lost the 20 people and slowed the other 60 down.

Since people currently cost more than computers do, it seems that such measures should be incorporated into evaluations of systems.

Figure 15

158 CAMBRIDGE BENCHMARKS (40 MIN.)

	WHEELER		RELEASE 3			
MEMORY	2 MEG	1 MEG	2 MEG		1 MEG	
RESPONSE	C.146	0.624	2.2	0.537	1.6	0.601
TRANSACTIONS	12.374	11.149	7.921	9.167	7.319	8.062
PROBLEM CPU	60.4%	50.3%	33.6%	37.1%	23.8%	43.5%
TOTAL CPU	95.5%	88.8%	66.6%	97.4%	51.2%	73.2%
DRUM I/O	95%	95%	65%	→		
DISK I/O	5%	5%	35%	→		
USERS	80	80	80	60	80	60

Figure 16 shows computer cost on the vertical axis and the service measure called the expansion factor on the horizontal axis. If the machine is underused, then most of its cost may be lost dollars. As the demand of computing services grow, the expansion factor also grows. At some point, the system is being optimally used. As the load builds beyond that point, much of the hardware usage goes into unproductive work.

Figure 16

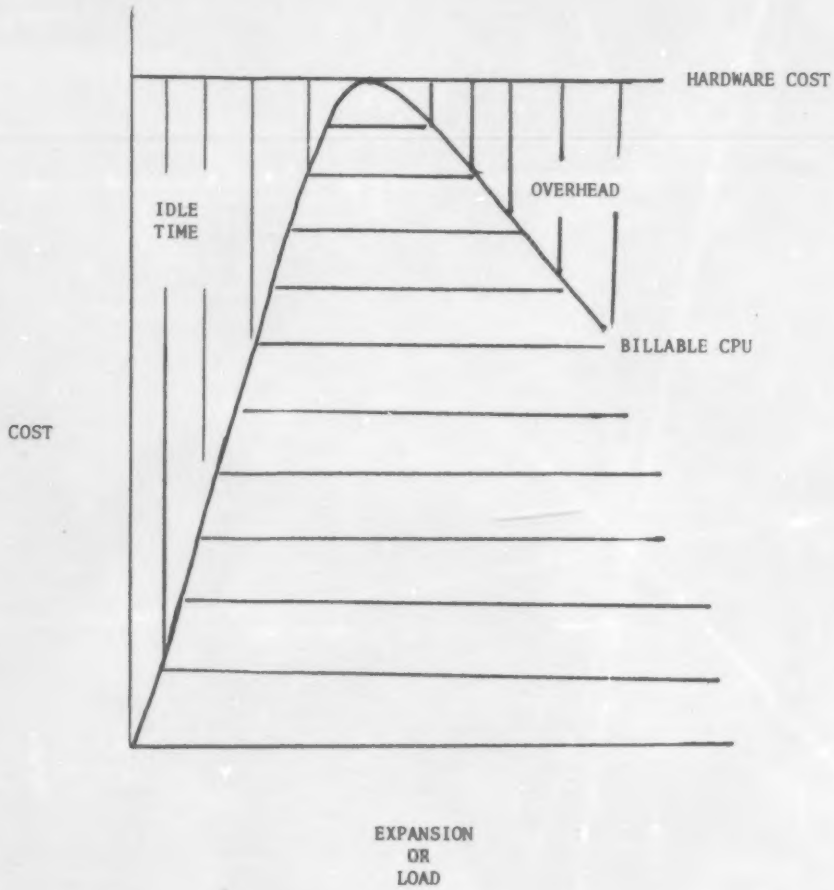
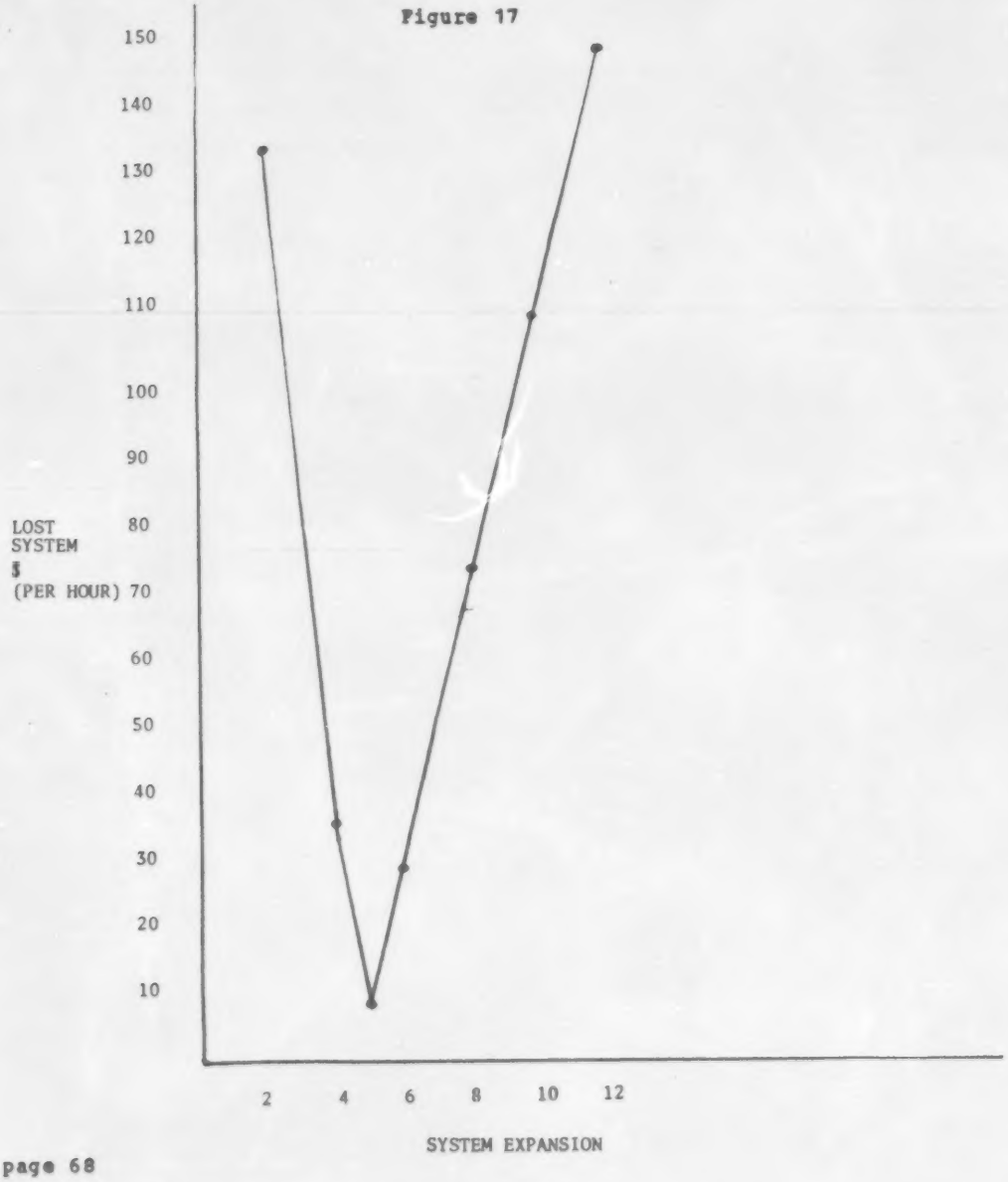


Figure 17 shows the lost hardware dollars per hour due to underload and overload on one of the VM/CMS systems at Yorktown. The V-shaped curve is an envelope containing the actual operating points. The expansion factor used here is an actual elapsed time expansion factor. It is the actual time to do some typical unit of computer-limited work divided by the minimum time to do that work in a stand-alone environment.

Figure 17



page 68

Figure 18 shows the cost of the lost end user time per hour on the vertical axis. The horizontal axis shows the expansion factor. As the expansion factor grows, the number of simultaneous users is also growing, and the number of simultaneous people being simultaneously slowed down by a slowly decreasing service to each also grows.

Figure 18

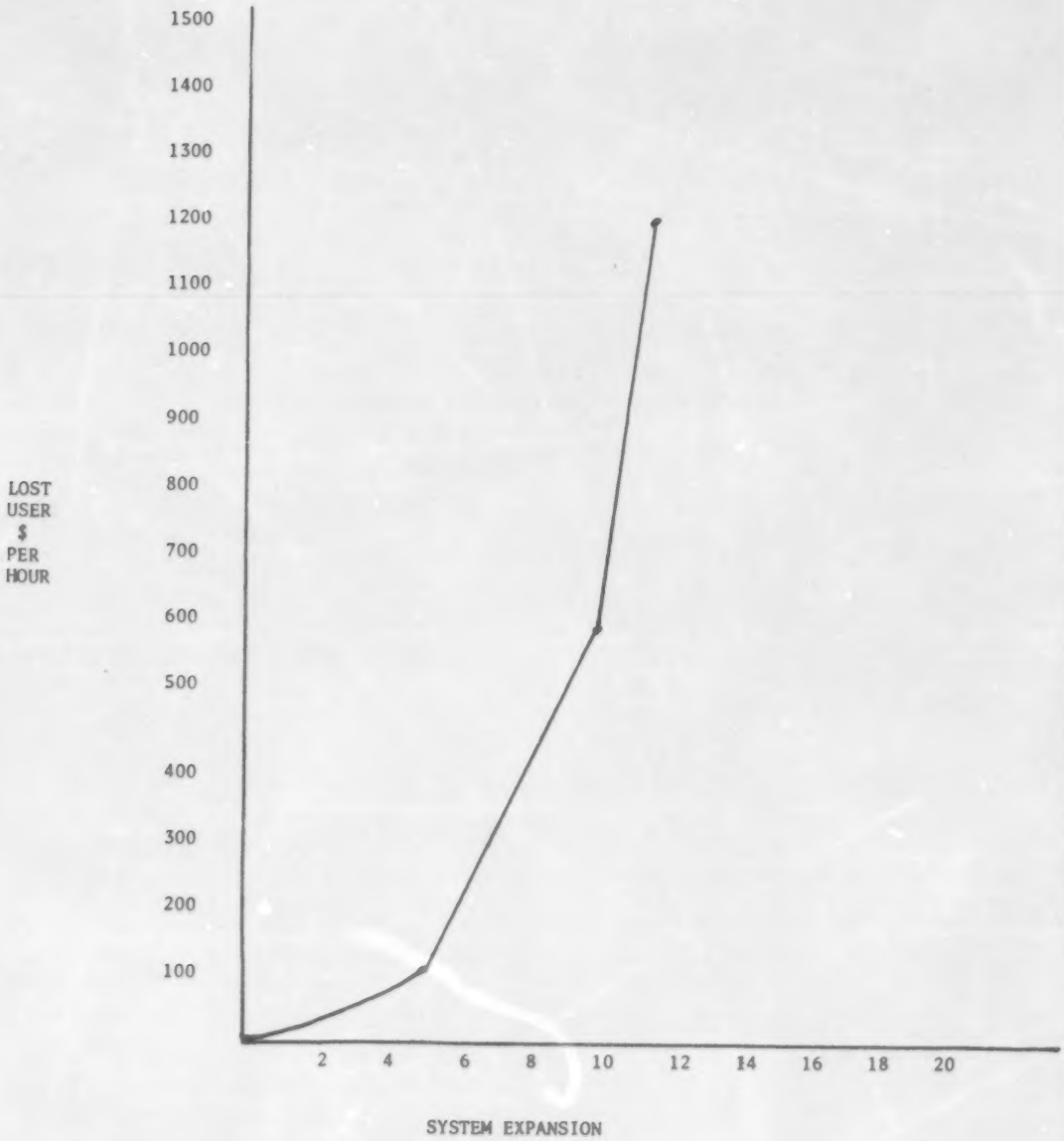


Figure 19 shows the combination of the hardware operating points and the end user costs as a combined loss. Note that there is indeeed an optimal operating point, having expansion factors in the range of 4 to 6. The cost of overload is far greater than the cost of underload when the end user's time is considered.

We use measures like these to monitor the daily use of our VM/CMS systems in Yorktown on our two 168s. We find that we can normally maintain the operating point very close to an expansion factor of 5 for most days.

Figure 19

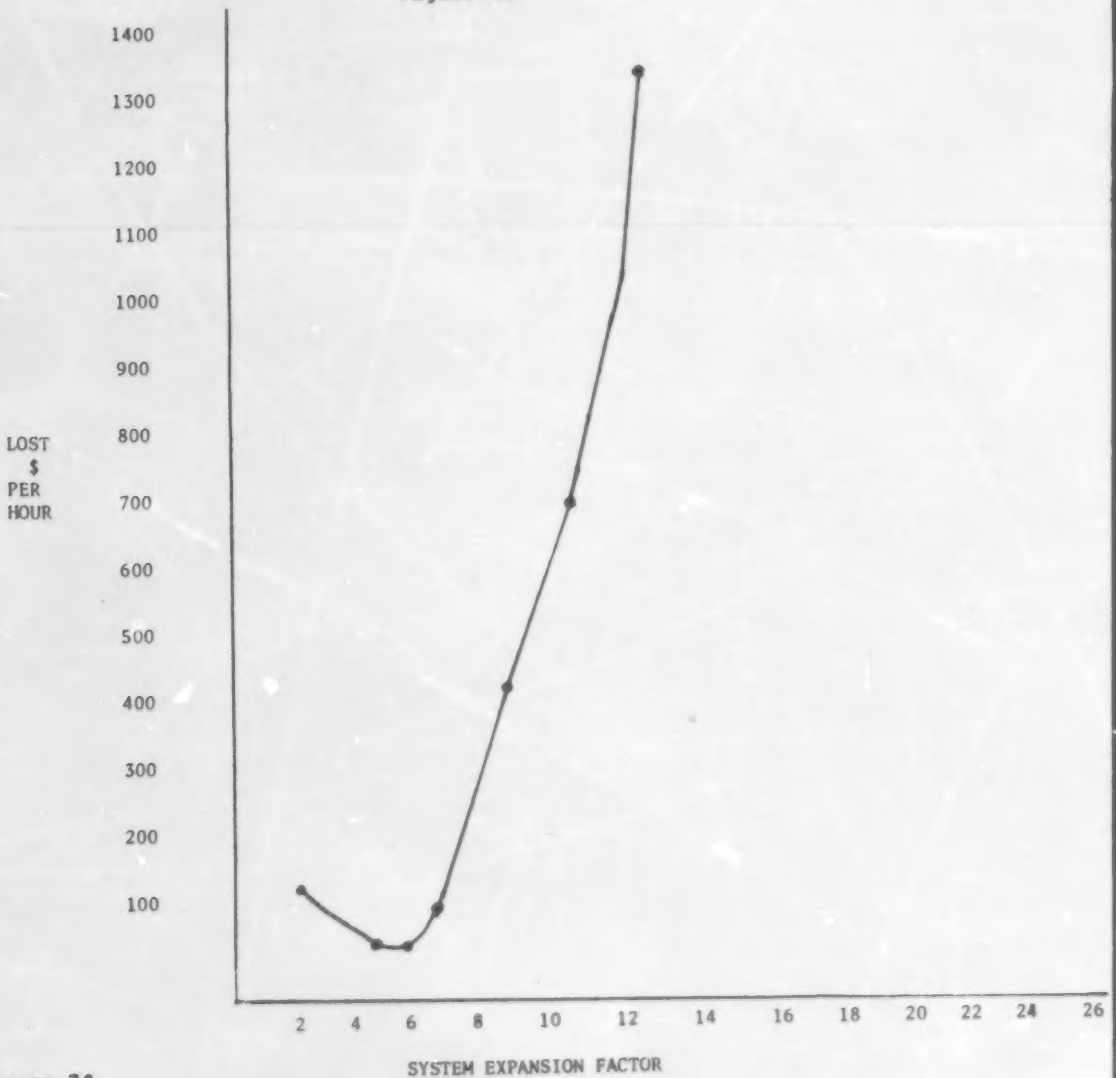


Figure 20 shows the five layers of storage in today's systems. Some of the problems addressed by managing each boundary in a better way are CPU speed, interactive response, virtual memory management and user growth limitation. By observation we have learned that data lives usefully in the cache for about 10 ms., in main storage for about 10 secs., on drum or fixed head file for about 10 minutes, on disk for about 10 days, and on archival storage, such as MSS, for many years.

The cache-main storage boundary has to do with CPU speed. Differences ranging from 2 to 1 up to 5 to 1 are achieved by proper cache management. This is because the locality of reference of most programs is such that a very small part of the storage receives 95 percent or more of the action.

The main storage - drum or fixed head file boundary has to do mainly with system response time. The issues here are the working set characteristics of programs to be run, the page replacement algorithms of the operating system, the parachor characteristics of the combination of these two, and management of the paging device channel program. Differences greater than 10 to 1 in interactive response time can be achieved by proper combinations of these four functions.

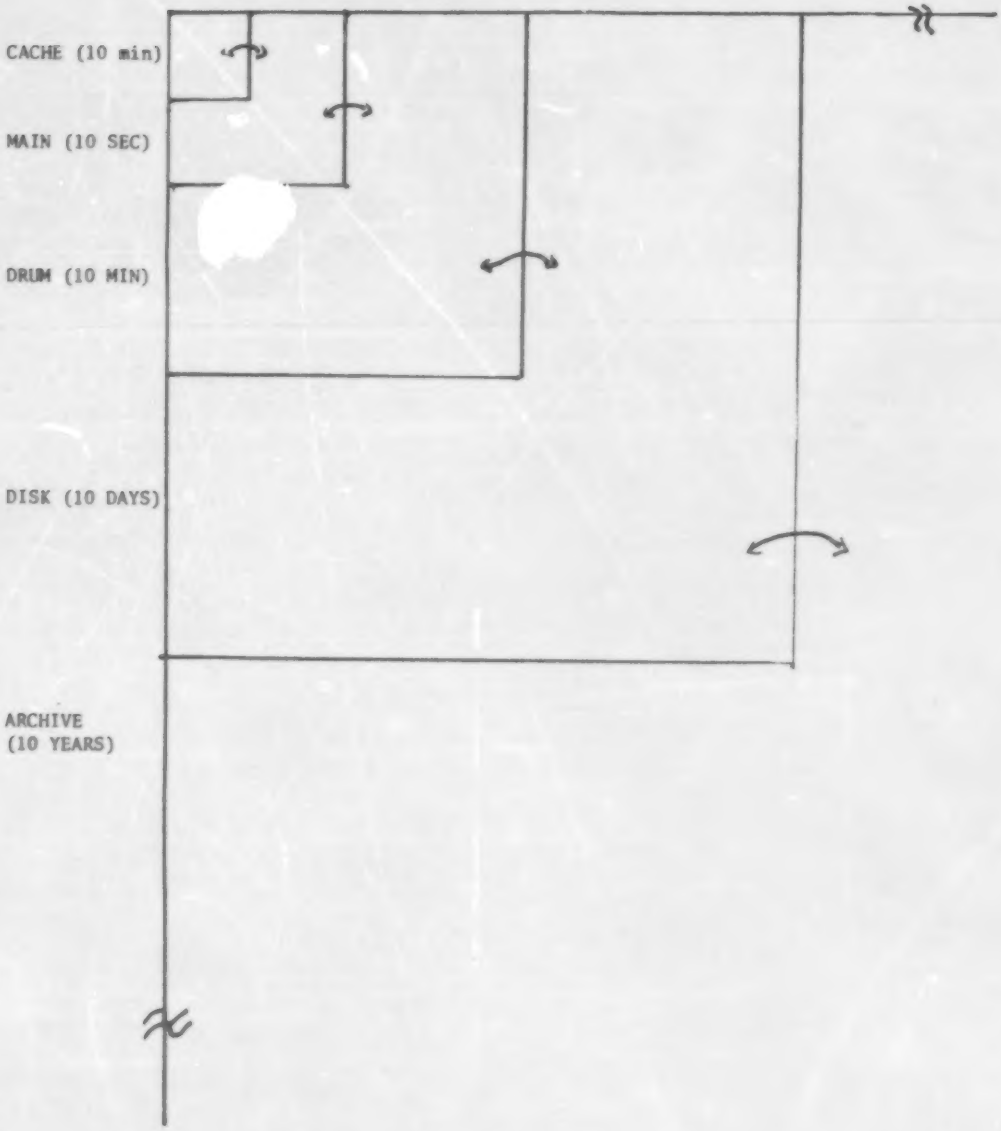
The drum - disk boundary has to do with the management of virtual storage. If there are 200 simulataneous users of the operating system, and each one has 2 megabytes of virtual storage, that is 600 megabytes of virtual storage. The 2305-2 fixed head files (drums) hold 11 megabytes of virtual storage on each drum. It clearly doesn't all fit on even 4 drums. However, the same locality of reference properties apply here as in all other levels of this hierarchy, only the time scale is on the order of ten minutes. Once again, differences ranging from 2 to 1 up to 5 to 1 are achievable by a good drum-disk migration scheme.

Finally, the boundary between online disk storage and online archival storage is the key to user growth and user productivity savings. Again the locality of reference properties apply and we find that 95 percent of online disk storage is never referenced for periods of time greater than a day. Furthermore, it has been found that users who fill their allocated online disk space spend a fair amount of time being data space managers from then on. The gating factor is not the amount of disk space they have. It is the users themselves. They inevitably free up just enough disk space to do what they have to each day. This is counterproductive on their part. However, the most important problem of all is the issue of user growth. By automating the disk - archive boundary, an open-ended secondary store effect is created. This opens up many new kinds of applications and a whole new way of using computers. At Yorktown the computing systems are treated as open-ended extensions of peoples; memory and reasoning power. This applies across the entire spectrum of tasks they do. It includes a wide spread of people.

Figure 20

STORAGE HIERARCHY

BOUNDARY PROBLEMS



The previous figures provide insight to the barriers and problems we have found and corrected. Motivated by what we have learned in terms of placing people productivity ahead of machine productivity, we have realized that the need for asynchronous processing becomes very much stronger. If people are to grow in the use of computers to extend their memory and reasoning power, then obstacles in the path of that growth must be removed.

At Yorktown we use 5100s and the IBM 7406 for many lab automation problems. We built TPVM to interface System 7s for bigger lab automation problems. Computer networking is used extensively to interconnect many different computing systems within IBM. In this way we can sharply reduce most conversion problems since we can nearly always find a site to run old programs. The message of "communicate--don't convert" is powerful.

We have developed many special function virtual machines to address those functions which we found to be very common. In this way the special function evolves in an environment which is uncontaminated by other non-related functions. This sharply reduces complexity while simultaneously allowing the user to do other work at his or her terminal as the asynchronous processing of the special function goes on.

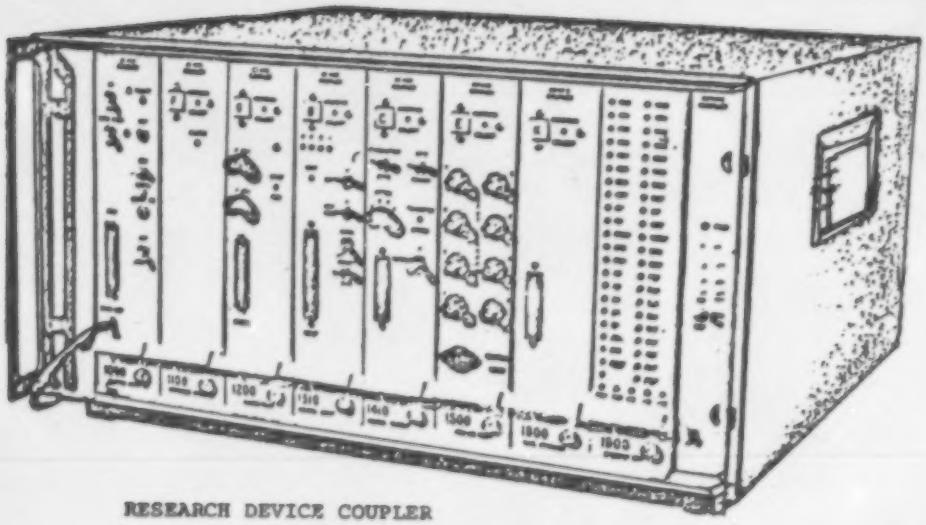
Some examples of this special purpose function are:

LAB AUTOMATION	5100 - 7506 (DEVICE COUPLER)
LAB AUTOMATION	SYSTEM 7s TPVM
NETWORKING	VNET (RSCS)
NETWORKING	DATA STAGE
TEXT PROCESSING	TTP
INFORMATION	BULLETIN BOARD
INFORMATION	NATURAL LANGUAGE INQUIRY
MACHINE TOOLING	VS1 - APT
SPEECH FILING SYSTEM	SFS
DATA SPACE MANAGEMENT	MIGRATE - ARCHIVE
CMS BATCH	NEW BATCH MONITOR
DATA SPACE MANAGEMENT	CPDIR
LIBRARIES	19D - 19E - 19F

The rate at which our users increase their ability to work in this environment is most impressive. They first show improved quality of what they do. Second they show improved quantity. In general the quantity shows an increase in the neighborhood of a factor of 10 in lab automation, document production, and program development.

Figures 21, 22, and 23 show a picture of the research device coupler and several ways in which it can be interfaced to the systems. The RDC which acts with, or as, a terminal, brings digital and analog input and output facilities to interactive terminal-oriented systems or to stand-alone systems such as the IBM 5100 Portable Computer. It is designed to provide those experiments with modest requirements access to high level language facilities, such as APL, FORTRAN, BASIC, PL/1, etc., for direct experiment control. This accounts for 70 to 80 percent of lab experiments. It eliminates the complexities of learning other operating systems.

Figure 21



RESEARCH DEVICE COUPLER

Figure 22

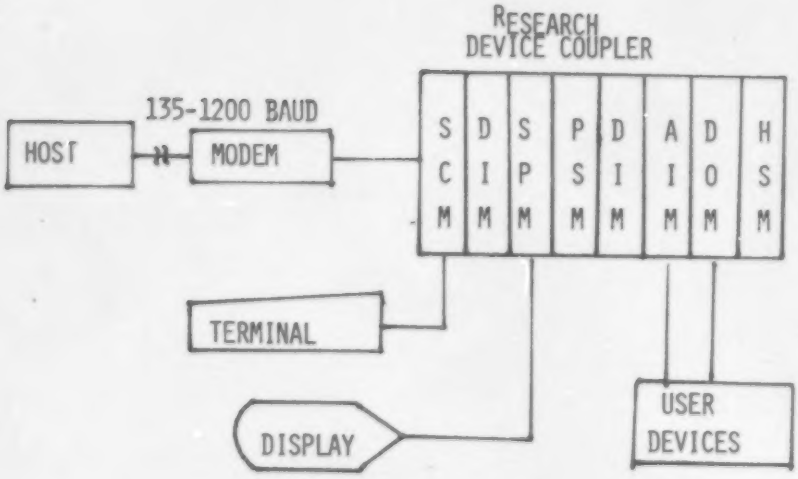


Figure 23

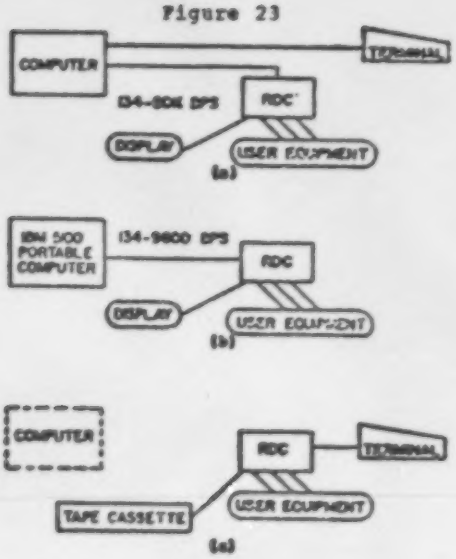
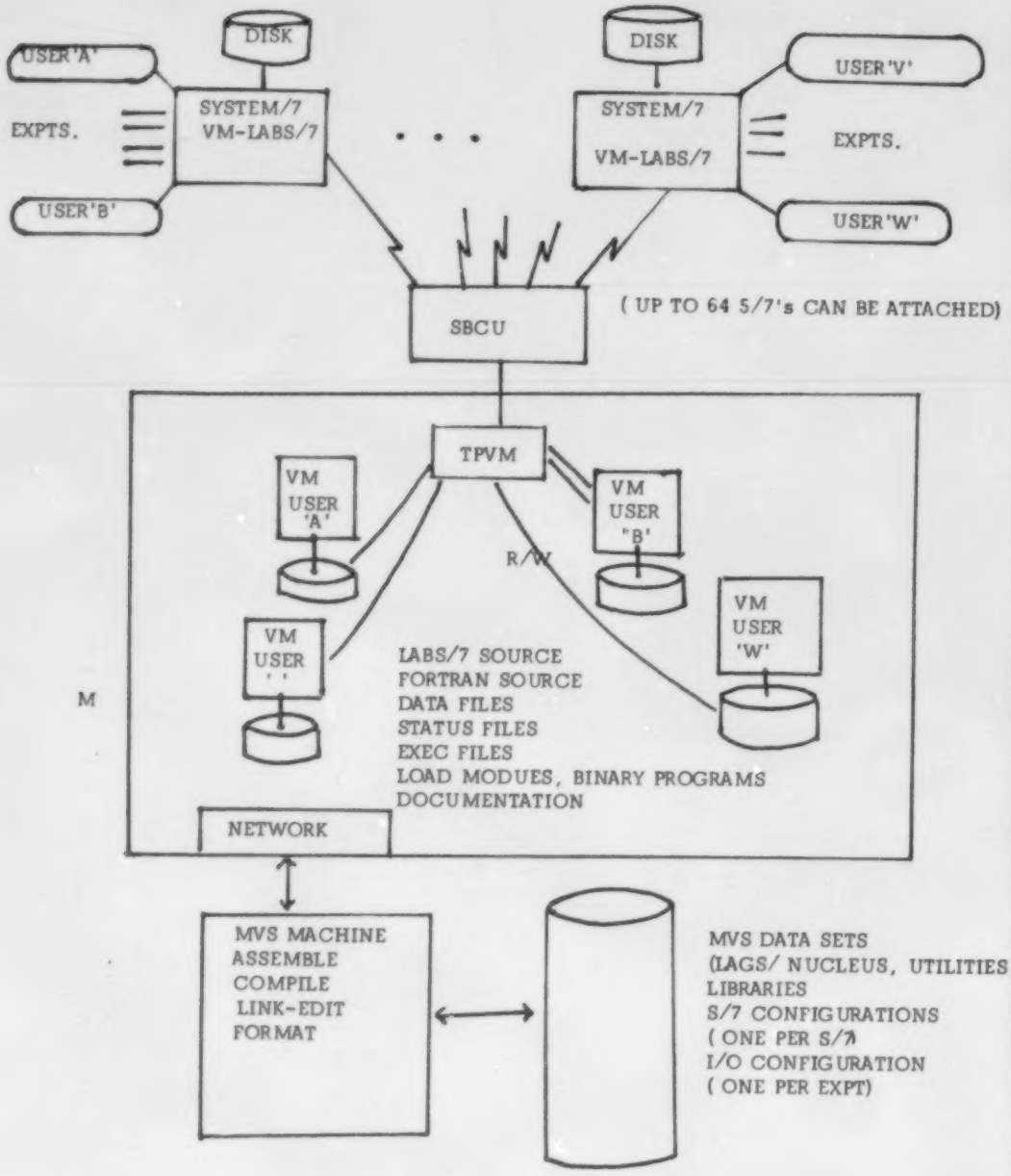


Figure 24 shows some of the distributed function which we have in place today.

The lab scientist, using Labs/7 (EDX) as the operating system in his System 7 can control multiple experiments. When the need arises to use a more powerful computing system either for processing or for accessing a data base, a simple mechanism called TPVM provides remote intelligent sub-systems with the ability to access and utilize the power of the IBM VM/370 environment. The scientist can directly access and edit data in VM/CMS from his terminal on the System 7. He can also initiate asynchronous processing of data either totally independent from what he is doing, or in a way whereby he can monitor the progress of the number crunching while doing other things. The speed of the interfaces are such that we have seen response times ranging from 7 to 40 ms. for the user at the system 7 terminal.

As the need arose to interconnect the different computing systems with different operating systems within IBM, the Subsystems Unified Network (SUN) was developed. This has recently become a series of IBM products, known collectively as NJI and NJE. We now have the ability for users of more than 190 internal IBM systems to communicate data to each other, run jobs remotely, and disseminate the results to as many diverse systems and terminals as they need to.

Figure 24



page 76

By having locally attached display devices as terminals, our users can normally achieve sub-second response time for almost all that they do. At the same time, they can communicate results, interchange technical documents, and freely communicate with colleagues all over the world. This communication takes place asynchronously from whatever they are doing at the terminal. A special function virtual machine manages the networking in our VM/CMS systems. Currently, VM/CMS systems constitute just over two-thirds of the systems in our SUN network. At Yorktown, about 500 people use the network to send and receive about 17,000 files amongst themselves and their colleagues on other systems each week.

The message "communicate data, not people" is powerful for interactive computing where the sub-second response time requirements are so strong.

Figures 25 and 26 show the early growth of the SUN network.

We have seen in earlier studies that editing accounts for 50 to 80 percent of what people do at terminals. One of the uses for editing is text processing. A special purpose virtual machine called TTF handles our text processing requirements at Yorktown. It provides the user with many useful facilities for composing documents. SCRIPT/370 is the primary formatting language used by our community. We have a compiler called STC (SCRIPT to TERMTEXT converter) which gives the user access to the more powerful TERMTEXT facitilies, including over 400 separate fonts at Yorktown computing center. Figures 27 and 28 show the sharp growth of the TTF service machine at Yorktown over the past year.

In summary, we have realized that the user at the terminal needs distributed function. In some cases this means special purpose virtual machines. In others it means interconnected computing systems. While in still others it means independently operating computing systems which may only be interconnected occasionally

We have learned that users steadily and rapidly increase the number of diverse functions they perform and that the requirements for better asynchronous processing grow proportionally to the interactive growth.

For the past ten years we have studied the way our people used interactive systems and observed barriers they encountered. They have been described in this presentation. The resulting working environment we have created for our Yorktown Research laboratory is highly productive and exciting.

Computer availability must be targeted for 24 hours per day and 7 days per week. If information is not available from a computing system when it is needed, then people do not keep all their data online. This implies that maintenance will have to be done time-shared.

Computer response time must usually be sub-second. This directly impacts people performance response time delays can lead to people delays, irritation, and more mistakes by people. Sub-second computer response time also opens up broad new areas of applications.

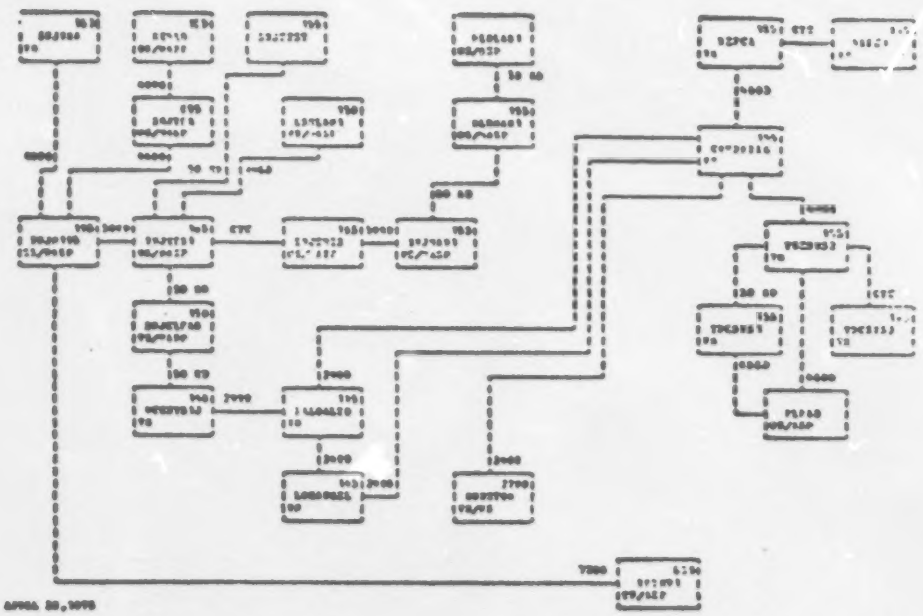
Data space management barriers, especially across main-drum, drum-disk, and disk-archive must be minimized. They are major barriers to economics of service and limits to user growth.

Complexity is the major problem facing the user today. By the use of EXECs we can sharply reduce complexity while increasing function. Also by using virtual machines for special purpose function, we minimize contamination of the function, thus reducing complexity.

The VM/CMS systems at Yorktown are the major interactive interface to all forms of computing for the Yorktown user community.

Today, over 25 percent of our laboratory is working interactively at any time throughout the day. Over 75 percent of our laboratory works interactively each week. There has been a factor of 8 growth in the time our people spend working interactively in the past four years alone. Over 98,000 hours of 'connect' time is measured in a month today. The users are present, and subject to system influence at least 38,000 hours per month, and at most 64,000 hours per month. Most of the remainder is for the special function virtual machines.

Figure 25



SUN NETWORK APRIL 1975

Figure 26

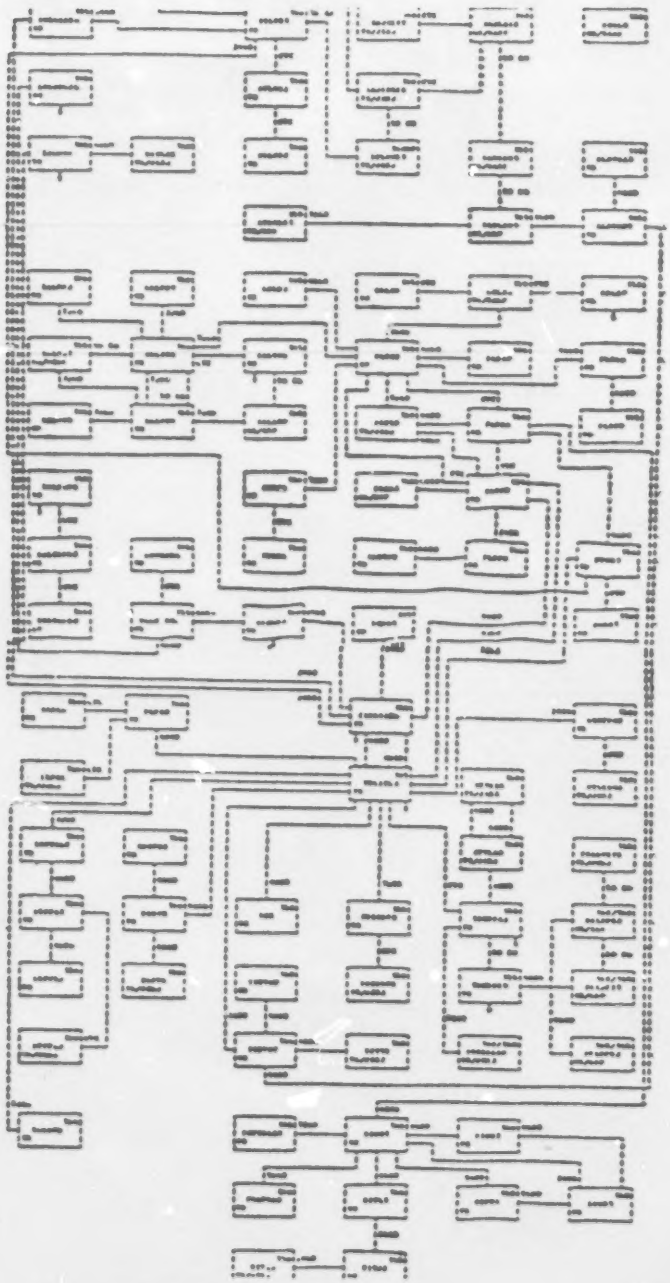


Figure 27

TTF SERVICE MACHINE SUMMARY USAGE STATISTICS

1976

	August	September	October	November	December	January
Total Monthly Job Counts						
Total	3054	3494	3854	3652	3724	4354
TTEXT	2454	2688	3025	3087	2981	3579
Non-TTEXT	600	806	829	565	743	775
First Shift	2568	2739	2926	2854	2999	3694
Normal Workday Job Counts						
Total	132	161	184	176	171	199
TTEXT	107	124	145	151	139	163
Non-TTEXT	26	37	39	26	32	36
First Shift	114	127	141	138	139	170
First Shift Small Jobs	95	102	112	113	112	134
First Shift Large Jobs	20	25	29	25	27	36
Percent Jobs From						
V	66	67	56	52	46	43
T	22	23	32	38	40	46
M	9	8	11	9	14	12
Percent Jobs Run On						
T	100	81	92	62	100	100
V	0	19	8	38	0	0
User Counts						
Total	152	185	190	193	175	186
TTEXT	128	150	154	153	146	156
Non-TTEXT	47	51	60	70	54	59
Normal Workday Total	40	48	54	56	52	59
Normal Workday TTEXT	35	41	45	50	44	52
Normal Workday Non-TTEXT	6	9	10	8	9	9
Monthly CPU Minutes						
Total	403	452	538	641	620	660
TTEXT	379	422	498	611	582	620
Non-TTEXT	24	31	39	29	38	40
First Shift	331	339	409	456	479	557
Normal Workday CPU Minutes						
Total	18	21	26	31	29	30
TTEXT	17	20	24	30	27	28
Non-TTEXT	1	1	2	1	2	2
First Shift	15	16	20	22	22	26
First Shift Small Jobs	9	9	11	12	12	12
First Shift Large Jobs	6	7	9	11	11	13
Turnaround Minutes						
First Shift Small Jobs	21	27	26	28	16	12
First Shift Large Jobs	22	26	30	54	36	23
DRAFT vs. FINAL						
Percent DRAFT (TTEXT only)	69	58	64	67	61	61
Percent FINAL (TTEXT only)	31	40	35	32	37	36
Total Monthly DRAFT	1688	1557	1922	2060	1826	2172
Total Monthly FINAL	1356	1880	1870	1551	1834	2063
Normal Workday DRAFT	73	71	91	101	84	97
Normal Workday FINAL	59	87	90	73	83	96
RTC vs. STC (TTEXT only)						
Percent RTC	73	68	64	52	49	47
Percent STC	24	28	32	43	47	49
LATE Option Usage						
Total	25	98	85	151	170	84
Normal Workday Percent	1	4	3	5	6	2

Figure 28

	1976 December	1977 January	February	March	April	May
Total Monthly Job Counts						
Total	3724	4354	5454	5927	4866	5200
TTEXT	2981	3579	4707	5301	4043	4608
FINAL Picture Forwarding	743	738	724	590	790	557
First Shift	2999	3694	4238	4602	4100	4364
Normal Workday Job Counts						
Total	171	199	267	249	230	239
TTEXT	139	163	231	223	194	213
FINAL Picture Forwarding	32	34	35	24	34	25
First Shift	139	170	207	194	197	202
First Shift Small Jobs	112	134	172	157	157	165
First Shift Large Jobs	27	36	36	38	39	38
Percent Jobs From						
V	46	43	49	45	43	47
T	40	46	45	48	44	45
M	14	12	6	7	13	8
Percent Jobs Run On						
T	100	100	98	100	100	100
V	0	0	2	0	0	0
User Counts						
Total	175	186	206	232	230	254
TTEXT	146	156	169	202	196	222
FINAL Picture Forwarding	54	52	57	54	65	57
Normal Workday Total	52	59	66	65	70	73
Normal Workday TTEXT	44	52	59	60	62	66
Normal Workday FINAL Forwarding	9	9	9	7	10	8
Monthly CPU Minutes						
Total	620	660	615	753	606	721
TTEXT	582	620	581	715	560	692
FINAL Picture Forwarding	38	27	26	22	38	23
First Shift	479	557	484	613	522	598
Normal Workday CPU Minutes						
Total	29	30	30	32	28	33
TTEXT	27	28	29	30	26	32
FINAL Picture Forwarding	2	1	1	1	2	1
First Shift	22	26	24	26	25	28
First Shift Small Jobs	12	12	14	15	14	16
First Shift Large Jobs	11	13	9	11	10	12
Turnaround Minutes						
First Shift Small Jobs	16	12	16	10	9	14
First Shift Large Jobs	36	23	20	15	13	26
DRAFT vs. FINAL						
Percent DRAFT (TTEXT only)	61	61	65	62	67	62
Percent FINAL (TTEXT only)	37	36	30	36	31	35
Total Monthly DRAFT	1826	2172	3041	3280	2691	2842
Total Monthly FINAL	1834	2063	2160	2514	2037	2161
Normal Workday DRAFT	84	97	150	137	128	130
Normal Workday FINAL	83	96	107	106	95	100
RTC vs. STC (TTEXT only)						
Percent RTC	49	47	45	44	35	33
Percent STC	47	49	51	55	61	64
LATE Option Usage						
Total	170	136	344	738	129	153
Normal Workday Percent	6	4	8	15	3	4

THE CASE FOR THE NETWORK

by

Thomas E. Kurtz
Kiewit Computing Center
Dartmouth College
Hanover, New Hampshire 03755
(603) 646-2923

Introduction

Despite our looking to the future as a purpose of this conference, in this gathering I cannot help recalling the past. CCUC/1 was held in this building -- in this room -- eight years ago, in June of 1970. Dare I say that it was then a new "concept". While many persons were involved in the planning of this conference, as well as in other national networking projects, no one worked harder and with more humor and grace than did Gerry Weeg. We are indebted.

When Ted Sjoerdsma was negotiating with me for a topic and title, he reeled off, as I recall, talk after talk about the wonders of microconputers. "They would do this, and they would do that, and finally the revolution in instructional computing was at last going to occur," he seemed to imply. So it was with some fear and daring that I elected to "defend" the old guard amidst the revolutionary onslaught, and to say a word or two about networking.

Now that the conference is half over, I see that my fear was misplaced, and my daring not needed. As I look out at you, the audience, I don't see the angry avant garde, I see my old friends, most of whom were here eight years ago when we were the avant garde.

Micros, to be sure, are revolutionary and are allowing us all to do the wondrous things that only a few could afford a few years ago. They are everywhere -- in our watches, our pocket calculators, and in our homes. Pretty soon we will all own one, if we don't already.

But it is also clear that change rates do not change rapidly. Even the 800 terminal plan at Berkeley (mentioned by Arthur Luehrmann) calls for the use of conventional timesharing systems.

Someone asked yesterday "Is timesharing dead?" That reminded me of a similar public pronouncement I made about seven or eight years ago: "Batch is dead." Without fussing about the definitions of "batch" and "timesharing", I think we can safely conclude that my remark, forceful though it may have been, had little effect on the sales of "batch" computers around the country. And I would estimate that any of us here who do obtain micros for experimentation or student use will not give up our own terminals to the larger system for some time to come.

What we extolled for timesharing ten years ago is now, at last, becoming a reality. Listening to Walter Doherty's presentation, we realized that even IBM has at last discovered interactive computing. What super interactive facilities, including a store-and-forward voice message system! I am green with envy at their worldwide network. Why shouldn't we in academia have at least a pygmy imitation of it?

I believe in the study of history. It suggests how things will go, and how fast, and helps us understand what we see. The glowing ads we see from Radio Shack for the TRS-80 remind me of glowing claims we all made for computing 15 to 20 years ago. Tom Lehrer once lamented his own lack of accomplishment in comparison with that of Mozart: "It is sobering to realize," Lehrer stated, "that when Mozart was my age, he had been dead for ten years." The level of BASIC now appearing in some micros has, at Dartmouth, been dead for ten years.

The ads remind me of my early experiences with the Datatron 205, the IBM 650, and the LGP-30. The true change between those machines and the current micros is not in their capability, but in their price. Their functional capability is a decade old.

Steve Hedetniemi reminded us that computer costs are now 10% hardware and 90% software. For $1000 you can get a lot of hardware, but not much good software. As Doherty said, we are being driven by hardware.

What I would like to do is to stand back, examine a model that represents the changes in computing, and express a hope.

A Network View of Computing

As I see it, apart from economics, which is critically important, the main issue is the <u>functional</u> architecture of the computing system. Of little or no importance is whether core or MOS is used for memory, whether slow or fast disk, whether LSI or whatever, or whether landline or microwave or satellite. The main issue to me is whether or not I can do what I need to do. If I can, it probably doesn't matter whether it costs a penny or a dollar; if it really serves my needs, I'll buy it. And much, no, <u>most</u> of what I want to do involves <u>communications</u>: communicating with colleagues, sharing files, transporting programs, accessing data bases that I do not own, and so on. Admittedly, some of what I want to do, such as accounting and occasionally calculating, does not require a network. But even text

processing, which would seem not to require a network, for me does require a network, since I want to share drafts of my "papers" with colleagues.

Figure 1 represents the state of affairs circa 1955-1960 shortly after primitive batch (punched card) machines became common. In the figures that follow, squares stand for machines of various sizes, dots stand for users, and dashed lines stand for the sidewalks that led to and from computer centers. Machines, being scarce, large, and expensive, were the masters. Only a few dedicated researchers and data processors used them. Student use occurred at night, if at all. Primitive sharing was possible, as shown by the institution on the right in the figure, though most early efforts were motivated by direct cost savings rather than by the benefits of intellectual sharing.

Figure 1

Computing circa 1955-1960

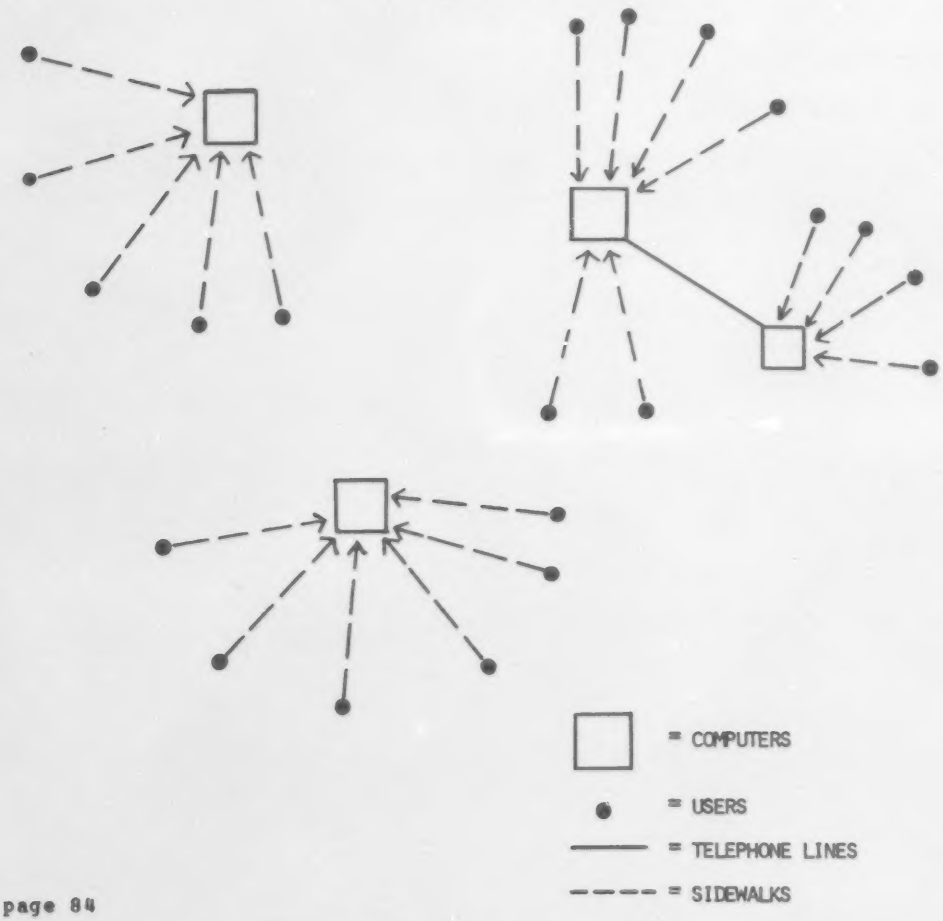
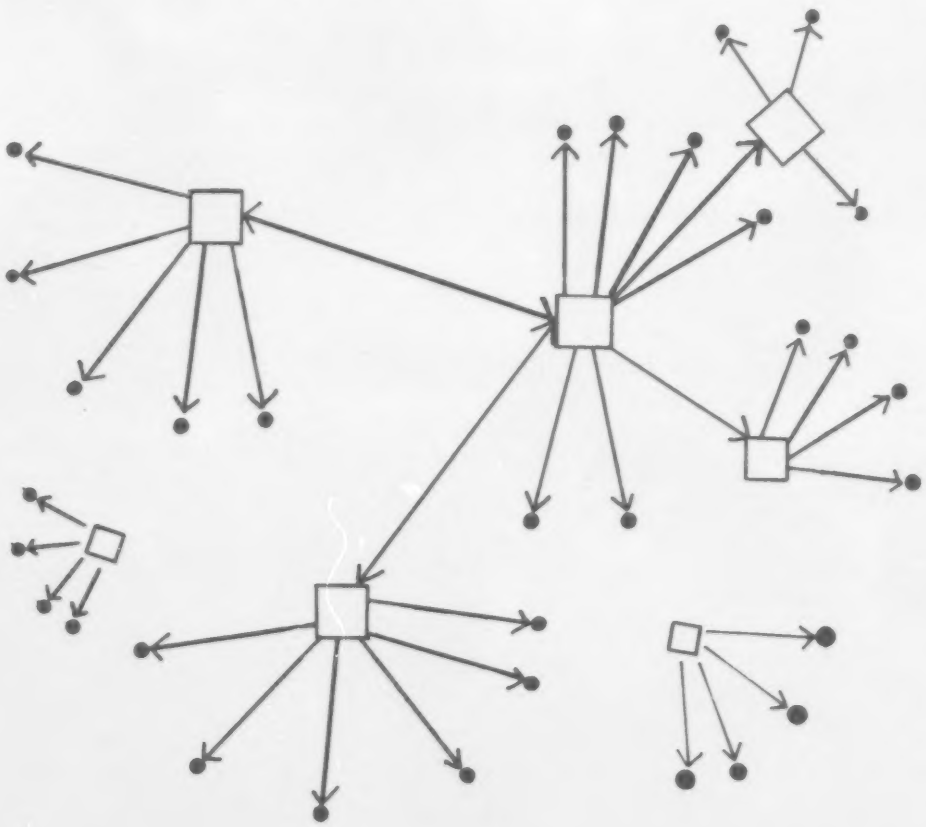


Figure 2, circa 1960-1970, shows the genesis of timesharing, which brought still-expensive computing to a larger audience. The era saw the development of mini-timesharing systems to supplement the big machines; these tended to land in the smaller institutions which lacked both a computer and an entrenched user community. Ironically, many of these smaller institutions were thereby able to do better at student computing than their larger cousins. The big institutions are only recently developing campus-wide (i.e., all students) plans for student computing and computer literacy. The important point about the situtation of Figure 2 is that the natural networking that occurred serendipitously produced something else -- a new level of intellectual exchange between students and scholars.

Figure 2

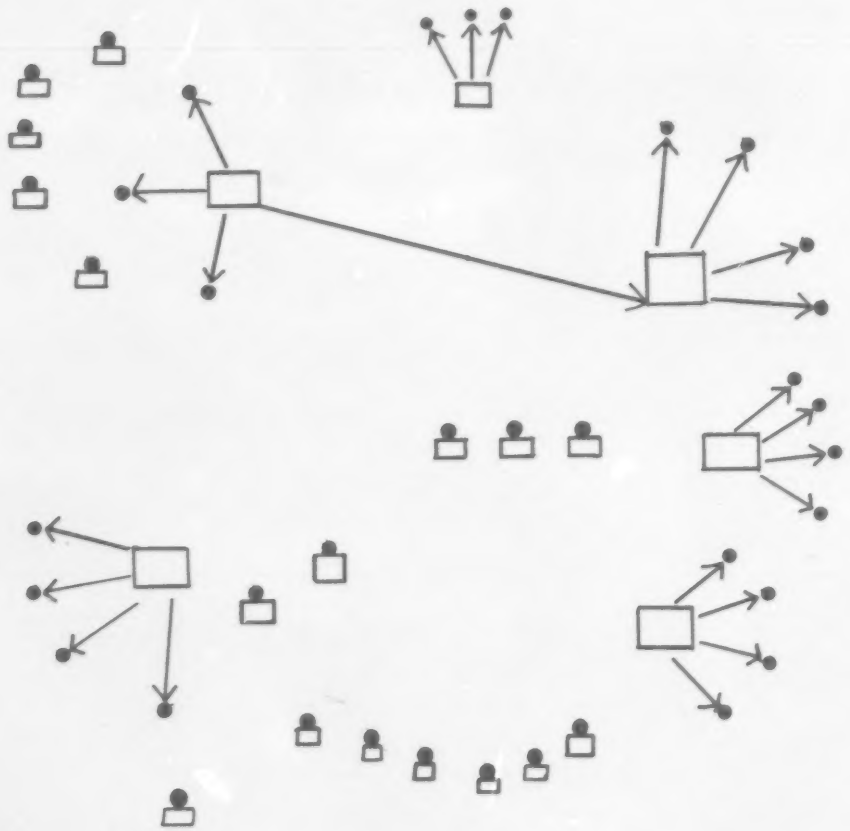
Computing circa 1960-1970



With microelectronics a new force enters the picture. Many services that forced us to go to expensive big machines or timesharing systems can now be had for a fraction of the cost. Simple student computing, teaching elementary programming, text editing -- all can now be done on individual machines such as the Radio Shack TRS-80 or the Commodore PET. This could lead to the situation represented by Figure 3. A chicken in every pot and two computers in every den. Which is a good thing -- computers take less energy than automobiles.

Figure 3

Microcomputers without Networks: The Future?

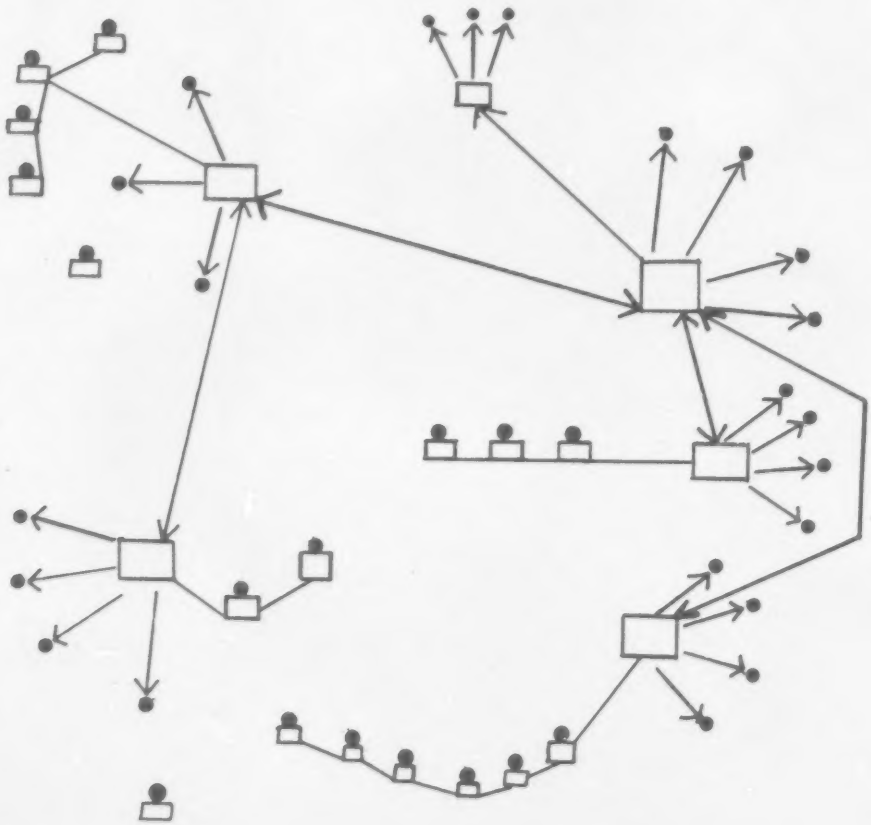


But one thing is lacking: communication. The stuff that links us together intellectually, now that we no longer need it to distribute routine computer resources, has disappeared with the baby that got thrown out with the bath water.

Figure 4 shows exactly the same situation but with the communications back in place. Please understand that lines represent functional communication, and I neither specify nor care what the actual physical mechanism is. The important thing is that one can now engage in joint work with one's colleagues, and can now use the special stuff that one's own machine lacks.

Figure 4

Microcomputers with Networks: The Future!!



Examples

Let me now mention in detail several activities carried out on computers where the larger network is required, or, at the very least, is beneficial.

1) Joint work on documents

My first example involves sharing and exchange of files and papers. One of my colleagues prepared a draft of a proposal which we planned to bring before our department. Seeking my suggestions, he notified me of the name of the file containing the draft. I merely obtained a copy of the proposal at my home, which is eight miles from his, generated several changes, and electronically mailed my responses to him. He revised the proposal the next morning, and before noon had listed sufficient copies for the entire department. Without the network, we would either have had to delay the presentation to the department, foregone the luxury of my suggestions, or have had a harried time with the department secretaries at the last minute.

2) Distribution of programs

Another example is the dissemination of programs. Industry now does this regularly; universities do not, but they should. CONDUIT program sets could be shipped by wire as easily as by tape. The conversions could be more simply made using a network. If the programs originally transmitted did not work properly, revisions could be easily made at the source, and the new versions transmitted. The time required for installing new program sets could thus be reduced from weeks to days, or even overnight, if the programmers at each end could work jointly through a network. DTSS Incorporated, a small software firm located in Hanover, ships some of its updated versions over the wire.

3) Using program libraries

Public libraries of shared programs are obviously easier to share and to maintain over a network. Some of the more stable and better maintained collections can be distributed by tape, cassette, or floppy disk. But most of the program collections, while having value and interest, fall short of being fully stable and fully maintained. These collections are best maintained by the author and made available over a network "as is". If the quality is good, they will be used. If the quality is poor, they won't be used, and no publication expense will have been incurred.

At Dartmouth, most of the departments have their own sublibraries. These are maintained by someone in the department, but are available to anyone on the network. It is thus relatively easy to encourage sharing of programs among, for instance, biology departments at different institutions.

Students can also share programs and other creations. At Dartmouth the sublibrary TRYITLIB******** offers free samples, and acts as a "hawkers and peddlers" free enterprise market place.

4) Classroom aids

As another example, lecturing aids tend to be large and complicated. In many cases, these should be shared by instructors within the institutions, or even in other institutions. This sharing can best take place using a network. Either the program can be transported, or it can merely be used on a distant computer.

5) Student faculty communication

Better communications with students is possible using an on-campus network. Electronic mail provides overnight question-and-answer service to students, in addition to whatever office hours have been scheduled. Assignments and memoranda can be stored in the course library in the computer, avoiding duplication and distribution of paper documents. The students might or might not produce a paper copy for their own notebooks.

Instructors can keep the course grades in the computer, even when the course is not self-paced. The students can at all times interrogate their own scores (but not those of their classmates) and can verify the correctness of the scores, thus reducing the possibility of clerical errors by the instructor.

A computerized signup service can be provided if the course uses scheduled interviews. A student can reschedule the interview at any time without having to locate the instructor to obtain permission. None of these course-related activities can be done on isolated computers; some form on on-campus network is required.

Although the instances of network applications described here should be familiar to most readers, the computerized signup service seems new. Two sample sessions are therefore included to illustrate the idea. Figure 5 shows a typical student session. The student first prints the array of available periods. He or she then signs up for period (2,3), and prints the schedule to verify the selection. If he wished to be reminded of his choice, he would have typed WHEN. He then has second thoughts and changes his choice to period (5,4). The previous selection is automatically erased.

Figure 6 shows the instructor's session. When he calls for a display, his students' user numbers indicate their choices. All X's indicate periods not available; in this case, the instructor had decided to schedule lunch on May 4 and 5. The instructor then decides to have lunch on May 6 and 7 as well; he adjusts the schedule by "blotting out" periods (4,3) and (4,4).

Figure 5

Student Signing Up for an Interview

```
 OLD #K52200:SCHEDULE
*RUN
To see the schedule, use the command PRINT.
To enter your user number into the master schedule, use
the command SIGNUP.  You may change your selection to any
other available period; your previous choice will then be
erased.  To find out when your period is, use WHEN.

ENTER SCHEDULE NAME? CONF


ENTER COMMAND? PRINT

Row and column numbers shown for available periods.

          May 4     May 5     May 6     May 7

9-10       1 , 1     1 , 2     1 , 3     1 , 4
10-11      2 , 1     2 , 2     2 , 3     2 , 4
11-12      3 , 1     3 , 2     3 , 3     3 , 4
12-1     XXXXXXXX  XXXXXXXX    4 , 3     4 , 4
1-2        5 , 1     5 , 2     5 , 3     5 , 4
2-3        6 , 1     6 , 2     6 , 3     6 , 4

ENTER COMMAND? SIGNUP

Enter row and column of choice? 2,3
Your choice has been entered.

ENTER COMMAND? PRINT

Row and column numbers shown for available periods.

          May 4     May 5     May 6     May 7

9-10       1 , 1     1 , 2     1 , 3     1 , 4
10-11      2 , 1     2 , 2    XXXXXXXX   2 , 4
11-12      3 , 1     3 , 2     3 , 3     3 , 4
12-1     XXXXXXXX  XXXXXXXX    4 , 3     4 , 4
1-2        5 , 1     5 , 2     5 , 3     5 , 4
2-3        6 , 1     6 , 2     6 , 3     6 , 4

ENTER COMMAND? SIGNUP

Enter row and column of choice? 5,4
Your choice has been entered.

ENTER COMMAND? PRINT

Row and column numbers shown for available periods.

          May 4     May 5     May 6     May 7

9-10       1 , 1     1 , 2     1 , 3     1 , 4
10-11      2 , 1     2 , 2     2 , 3     2 , 4
11-12      3 , 1     3 , 2     3 , 3     3 , 4
12-1     XXXXXXXX  XXXXXXXX    4 , 3     4 , 4
1-2        5 , 1     5 , 2     5 , 3    XXXXXXXX
2-3        6 , 1     6 , 2     6 , 3     6 , 4
```

Figure 6

Instructor Examining Signup Sheet

```
USER NUMBER--K52200,
*OLD SCHEDULE
*RUN
To see the schedule, use the command PRINT.
To enter your user number into the master schedule, use
the command SIGNUP.  You may change your selection to any
other available period; your previous choice will then be
erased.  To find out when your period is, use WHEN.

ENTER SCHEDULE NAME? CONF


ENTER COMMAND? PRINT

Row and column numbers shown for available periods.

          May 4      May 5      May 6      May 7

9-10       1 , 1      1 , 2      1 , 3      1 , 4
10-11      2 , 1      2 , 2      2 , 3      2 , 4
11-12      3 , 1      3 , 2      3 , 3      3 , 4
12-1      XXXXXXXX   XXXXXXXX    4 , 3      4 , 4
1-2        5 , 1      5 , 2      5 , 3     HD10284W
2-3        6 , 1      6 , 2      6 , 3      6 , 4

ENTER COMMAND? BLOT

Enter row and column to blot out? 4,3

ENTER COMMAND? BLOT

Enter row and column to blot out? 4,4

ENTER COMMAND? PRINT

Row and column numbers shown for available periods.

          May 4      May 5      May 6      May 7

9-10       1 , 1      1 , 2      1 , 3      1 , 4
10-11      2 , 1      2 , 2      2 , 3      2 , 4
11-12      3 , 1      3 , 2      3 , 3      3 , 4
12-1      XXXXXXXX   XXXXXXXX   XXXXXXXX   XXXXXXXX
1-2        5 , 1      5 , 2      5 , 3     HD10284W
2-3        6 , 1      6 , 2      6 , 3      6 , 4

ENTER COMMAND? STOP
```

6) Experimental use of new ideas

Since developments now appearing on personal computers first appeared ten or so years ago, we may postulate a ten-year lag from original development to general use. For example, the versions of BASIC that are now appearing on microcomputers are similar to Dartmouth BASIC of 1965-67. During the development decade, the new idea can be tried out with ever-increasing audiences through networks. An example is the development of BASIC itself. In 1971, Dartmouth introduced its sixth edition of BASIC, a version whose capabilities are just now beginning to creep into non-Dartmouth BASICs. After being stable for seven years, a new version that includes plotting and structured elements is being designed. Not yet stable enough to distribute as a finished product, users can easily try it out on the network and thereby both benefit from its advantages and contribute to its development.

7) Access to data bases

Yet another application is access to large data bases, a long-recognized benefit of networks. Most of the benefits accrue to administrative and research applications, but we must remember that today's research becomes tomorrow's courses. While student use of large data bases is not yet common, to deny such use by forcing all student applications to isolated computers would be unnatural.

8) Natural language processing

A final example is the new field of natural language processing. Even with the enormous price drop, it is unlikely that typical microcomputers will be able to support natural language interfaces to data bases. But the price drop is making it possible to consider such developments on computers large enough to sustain these developments. Such large programs or systems need to be shared to be effective to a larger group of users.

Summary

A large number of important applications of computing require that a geographically diverse group of users have access to a single program or data base through a network. Even if such applications constitute a minor portion of total computer use, they are nonetheless important applications and must not be denied.

What do we need to make sure that, as microcomputers develop, network connections also develop? First, gadgets are needed to allow microcomputers to be connected to telephone lines. In addition, software in the micros needs to support a simple and standard form of communication.

Second, the larger timeshared computers need to permit access not only from human users, but also from other computers. Additional software is needed to allow a connection from another computer, to "converse" with the other computer without using the human command interface, and tc permit at least file transfers.

Third, continued work must be done on specifying standard languages, especially control languages and communications protocols.

Fourth, we need more communications systems, such as TELENET and TYMNET, to support the interconnection of microcomputers. Perhaps PCNet is an example of one.

As a last word, business and industry have discovered networking and the better communication it allows. Colleges and universities appear to be discovering microcomputers. Let's not make the mistake of assuming that isolated micros, because they are cheap, deserve our complete attention. Don't let's put all our eggs into one basket.

THE MICROPROCESSING REVOLUTION AND ITS EFFECT ON SMALL COLLEGES

by

Sister Mary Kenneth Keller
Director of Computer Services
Clarke College
Dubuque, Iowa 52001
(319) 588-6391

One of the meanings of revolution, as given in a standard dictionary, is "an extensive or drastic change in a condition, method, or idea." Using this definition, we can say that in the past few years a microprocessing revolution has occurred. For there are, indeed, extensive, if not drastic, changes taking place in conditions, methods, and ideas associated with computing due to the increasing development of the microprocessor. In this paper we wish to assess the impact of this revolution, especially as it relates to the microcomputer, on small colleges.

A college of any size generally makes choices as to which needs and what types of students it can or should serve. A small college must make careful choices, since it is not in the position to satisfy the needs of everyone. However, no good college chooses to limit essential resources. To do so would reduce its effectiveness for any type of student. Long ago, small colleges became just as aware as larger institutions that a computer is a powerful tool for education.

The need for computer facilities has been solved by small colleges in various ways. One way was to join a network; another, to acquire a small, but limited, in-house computer; yet another was to do both. There were disadvantages to these solutions which can be summed up simply -- you got what you paid for. The small computer of the past decade had reduced capacities. Networks were not really cheap, and because of this, they rarely provided full computer usage on small campuses.

The first change the minicomputer is bringing is the presence of processors on every campus, and, most likely, more than one. For the same amount of money that was invested in a small-sized computer ten years ago, at least ten microcomputers of greater capacity and capability can be purchased now. Distributed processing is likely to be part of the educational scene, just as it is in the business world. Administrators in the past often took a dim view of acquiring multiple limited-capacity computers, especially if they were incapable of doing data processing. If there is still a justification for this attitude, there are now at least some arguments against it.

The following features are important to the use of a computer in education:

- an interactive mode of operation;
- with a display such as a CRT, a means for getting hard copy;
- sufficient memory to perform non-trivial tasks; and
- auxiliary storage such as disks and tapes with capacity to hold educational materials of a fairly comprehensive nature. materials of a fairly comprehensive nature.

The present day microprocessor-based systems can have these features at a very reasonable price. Because they are affordable, they may soon reduce, if not replace, timesharing.

One advantage for the student would be the actual contact with a computer, with an opportunity to explore and manipulate the computer as a system. In spite of our great strides in computer education, there is still a good deal of illiteracy about the computer as a system. This is not confined to persons who have had no experiences with a computer. To illustrate what I mean, I would like to repeat a conversation I had with an elementary school boy:

"Our principal is happy because we have a computer," he told me.

"What is a computer?" I asked.

With a great deal of assurance he replied, "It is a typewriter in a closet with a telephone."

Noticing my interest, he continued, "It calls other computers."

"What do they talk about?"

"Oh, they don't talk. They just type games at each other. The principal likes it."

As we all know, our perception of reality improves with experience, but don't count on it improving just with age!

In the first years of computer usage on small campuses, administrators often regarded the campus computer facility as a potential administrative data processor, and created contention for its use. First there was a creeping intrusion, then finally almost a total takeover of the facility by offices other than academic.

Students who might have initially explored the computer with long hours of curiosity were banished from the laboratory as it took on the aspects of a computer service bureau. The microprocessor brings the hardware back to the student. Microelectronics can be introduced as a field of study, even in laboratories with limited resources. There can be a greater association of software concepts with hardware than was possible before.

Meanwhile, administrative data processing, which is important for a small college, faces the prospect of better facilities than were ever available before. The same distributive processing that enhances the student facilities is available to administrators. Many colleges which before could not afford management information systems will find that the microprocessors provide them with convenient online systems. Moreover, experience with data management in their own offices will provide an education for management that more remote computing did not supply. This in itself will improve computer usage at all levels.

There is a growing tendency for small colleges to increase the number of courses in computer science being taught as a career-oriented sequence. There is a demand for this and it makes sense to provide it. There is a problem, however, which is the lack of computer professionals interested in and prepared for teaching. Major universities can help in this situation by providing more teachers of computer science. The number of businesses of medium and even small size who are automating their operations is growing by leaps and bounds because of the microprocessing revolution. Colleges of all sizes need to prepare more professionals to help customize programs for microprocessors for this growing market.

Essentially, what is happening is that there are now computer facilities for everyone. Many professions that formerly felt little need for computer professionals now have an urgent need. Small colleges that often supplied the needs in these professions need teachers with the necessary expertise to provide this training. Business and management majors need more insight to the newer technologies than were given to them in the past. We must realize that the small business man or woman will be more involved with computerized systems than might have been expected in the past. The small college need not feel that it is incapable of giving such training. It is more likely now that many businesses will be using the same systems that the small college is finding cost effective.

When the computer came upon the educational scene, there was great hope for its use as an instructional tool. Nevertheless, many faculty members who were initially enthusiastic about using a computer found it difficult to sustain this enthusiasm when well-designed pedagogical materials were not available for the system configuration to which they had access. Even today, after all these years of projects and programs, the potential of the computer in the educational process is scarcely realized.

The problem, as I see it, is somewhat like the one that affects many new users of computers in business. Once the salesman has demonstrated the software that is available, the system appears to be ready to meet the needs that gave rise to the selection of the hardware in the first place. But the snag is the very adaptability of the software package, which turns out to create a task for the adapter

requiring a large investment of time. A parallel situation exists in many educational packages.

With the availability of inexpensive but powerful microcomputers, the college teacher may find that he or she can invest more time in developing new or adapting old packages to current instruction simply because the computer is more accessible. We use typewriters and books in our offices to create instructional materials, and we need to have this convenience with respect to computer use. When this occurs in the offices and laboratories of smaller institutions, a whole new set of creative people will appear, with a concomitant rise in the production of computer-based instructional material. Designing software for education is an unfinished, or rather, a never-to-be-finished task, but it can be kept moving abreast of needs.

Maintenance of computer systems has been an expense and sometimes a great inconvenience for the small college. If we begin to build our own computers, as we may, then perhaps we can maintain them too. At least it should be cheaper to do so.

Lastly, the computer can come to class. It is a shame that we teach about software and hardware without the immediate presence and assistance of a computer. The small size and the portability of a microprocessor make it a good teaching assistant. It should be available in every classroom, just as a blackboard or a projector might be.

Will this distributed processing lead to the demise of networks in the educational field? I hope not. A computer is one of the greatest instruments of communication. The microprocessor itself is revolutionizing the communications field. Far from destroying networks, I look for a new and better designed network that will be more affordable and can be shared by institutions of all sizes.

THE RESHAPING OF MEN'S MINDS VIA COMPUTING

by

Grace C. Hertlein
Associate Professor
Department of Computer Science
California State University, Chico
Chico, California 95929
(916) 895-6438

Purpose

As computer science permeates almost every endeavor of man in technologically developed countries, it affects the human mental processes. In this conference we are addressing higher education and computing, yet the implications of computer use, and the resultant reshaping of men's minds, penetrates every educational level. I will address my remarks to the education world, but will remain aware of the practical world, since I believe in their interdependence. As higher education is charged with the responsibility of preparing people for professional roles in industry, education, government, etc., it is intimately related to the practical world of man's working life. The ideas presented in this paper will explore the specific objectives of this conference, but will also explore a larger view, that of the reshaping of men's minds via computing.

In order to use the computer, the minds of men are reshaped and massaged by the machines they use. As this practice of computing becomes more complex and more common, it is methodically altering, changing, extending the minds of men. Further, the products of computing -- the thought processes of men combined with the electronic processing capabilities of computers -- are changing, enhancing, reshaping the minds of men.

My purpose is to examine what this reshaping of men's minds via the computer means -- to give examples of mind enhancement, activated and energized by the use of computers. My purpose is to examine the changes in education and thought, and to anticipate further patterns of change.

Lower Level vs. Higher Level Thought

To go beyond rhetoric, the reshaping of men's minds via computing represents a pattern of absorption of the new mental power available to man, represented by the processing power of computers. It is an adaptation of the organism (man) absorbing a new mind potential (the machine), and results from responses to the new stimuli of computer systems and advanced technologies.

During the past twenty-five years, the methods by which many men think have been irrevocably altered and reshaped by computer technology. There is less "lower level processing" by man, such as sorting, comparing, storing, and retrieving data. These tasks have been assigned to the computer, to execute in millisecond precise performance. Today there is greater emphasis by man on more complex forms of mental activity: design of new systems; invention of new computer languages; formulation of new problems to input into increasingly more complex, more competent, and faster computers.

In allocating these "lower level" thought processes to computers, man has advanced beyond some of the mental processes that were formerly a part of his "thinking" activities. A greater emphasis can be found in the areas of logical algorithmic thinking because of the need to communicate with the computer, to solve problems with the aid of computer systems.

New Thought Patterns

At the present time, the computer calls for a special type of thought pattern development that is a by-product of the scientific method and mathematical processes. It is logical, painstakingly tedious, and proceeds in orderly precise steps or modules. I have often referred to the computer as the "discipline, or art, of minutia", because of these precise, minute requirements. This is not to imply that this pattern of thought is trivial, but rather that it proceeds in minute increments.

This pattern of thought is readily adapted to man's present known patterns of scientific thought. It does not require great "leaps of the imagination". The small steps required to communicate with the computer, however, are presently leading to larger steps, which in turn form newer larger patterns of thought by man as a species.

When this precise pattern of rational algorithmic thinking is combined with a second new thought pattern, the "what if" method, the reshaping of man's mind is enhanced. The second pattern exceeds the precise questioning of the scientific method.

This new "what if" pattern is a direct result of "massage" by the use of computers -- for the medium of the computer invites deliberate

exploration of innumerable alternatives in solving problems. Exploration of these diverse alternatives is not possible manually, because of time requirements and accuracy.

The "what if" pattern has now become an integral part of the computing man's present thinking patterns. The steady practice of these combined patterns, algorithmic thinking and the "what if" pattern, produces a steady stream of creative alternatives -- and resultant new inventions.

In essence the scientific method has been augmented by an abundance of alternatives and an abundance of processing power. This combined power of man and the computer is altering the shape and content of education, discovery, the arts -- the entire realm of knowledge, both theoretical and applied.

Comparisons: The Book vs. The Computer

To appreciate the reshaping of men's mind via computing, it is useful to recall the alteration and transformation that took place with the invention of the printing press. The availability of information via books reshaped the minds of men. The book reproduced in concise form the ideas of great minds, then disseminated this knowledge for reading and absorbing. More people could share the fruition of great minds via the book. The foundation of known knowledge was enhanced and augmented.

Yet the processing and dissemination of information via books resembles the image of a tortoise when compared with the jet-like speed of computers. For the computer not only gathers information, it compares, sorts, adds, augments, subtracts, rearranges information. The book could not perform this wider range of processing. Consider the speed of the printing press -- and compare the speed of the newest output devices of the computer!

The power of the computer to enhance the minds of men surpasses all other inventions of man to date. This machine is altering the way we think, the way we live, the way we do business, the way we communicate, the way we teach, even what we teach. It is enhancing and augmenting the intellectual energy and processing capacities of men. Just as the airplane gave wings to man's physical legs, the computer is giving wings to man's mind.

These are pleasant and stimulating ideas, but what do they mean specifically?

A New World to Explore

The territory of man's world of knowledge has been exponentially expanded by computing. The boundries of our known world of knowledge have been exploded -- there is an open unending space where known boundries previously existed.

Problems can now be solved that previously were too immense for the slowness of man's mind. An example is the research that was begun in Limits to Growth[3]. Never before in the history of man could human

beings afford the luxury of considering the "predicament of man". The problems of energy, natural resources, the population of the earth, food supplies, etc. -- all these represent immense data, complex variables, unsolvable by manual methods.

We now know that the computer's limits are our limits. As our skill grows, these limits are extended.

The world of knowledge in higher education is vastly greater because of the computer. The problems that students may study are more complex, but they are also more solvable, because of computer use.

A New Method of Questioning

At the present time, in order to use the computer, it is necessary to define a problem and its sequence of execution in an orderly manner. This may not be true in the near future. Many respected and well known people in computer science anticipate the "understanding creative computer". Nicholas Negoponte of MIT is one of these persons[4]. Edmund C. Berkeley, the founder of ACM, the author of Giant Brains, is another[5]. Thus the day of precise formulation of problem statements is perhaps an early phase of computer usage.

However, this need for precise formulation has been a boon for man's thought patterns. Man has learned of necessity to ask new questions, and to define precisely what was accomplished manually, and often, intuitively. Man now knows more about what he was and is doing. The tasks and aspirations of man are more measurable.

In addition, we have an enhanced method of asking questions. We have become more facile in definition, algorithmic thinking, and synthesis via computing.

A New Capacity of Process

In the past, if a scholar dedicated a total lifetime to problem-solving, one lifetime would be insufficient for many tasks. With the aid of the computer, the "thought lifetime" of man has been expanded to 10,000 years[6]. The power of the computer to process information at the speed of light has allowed man to process exponential quantities of information, and to process problems of exponential complexity.

The increased speed of the computer and the decreased cost of processing are known facts to every computer user. The future capacity of man and the computer to process information together represents a continuing reshaping of men's minds via computing.

The Extensive Expansion of Computer Applications

Wherever information is processed, the power of the computer has been and will be employed. "Prediction . . . every defined intellectual operation will be done faster and better and more reliably by a computer than by a human being."[8] The key word in the

last sentence is "defined". More tasks and problems and activities are being defined by man since he invented and applied computers.

In reviewing the applications of the computer, it appears that there is little that the computer has not touched. The "Computer Directory and Buyers' Guide", published by Berkeley Enterprises, Incorporated, listed 2600 specific computer applications in 1974. How many applications are there today? A prudent estimate, according to Mr. Berkeley, would be 5000.

Computer applications are task-oriented and discipline-oriented. In the former, specific software packages have been developed to accomplish certain jobs or tasks. The focus is on the task: using a statistical package on a computer library; drawing different types of maps; achieving a variety of charts and graphs from data; executing a series of drawings for the erection of a building.

Other applications are discipline-oriented: engineering, mathematics, physics, art, poetry, music, architecture, medicine, business, political science, geography, etc.

In analyzing the 2600 applications named by Berkeley, four main groups emerge:

1) Business and Manufacturing in General
2) Business -- Specific Fields
3) Science and Engineering
4) Humanities

Each main group is subdivided into an alphabetical listing of areas. Each area is subdivided into an alphabetical roster of applications. In area 1, for example, there are 121 office applications and 79 plant and production uses of computers.

Every person who uses computers in their work has experienced a reshaping of their mind. New working and thinking patterns have resulted in an enhanced mental capacity. With the growing use of computers in offices and institutions, literally millions of person have experienced a reshaping of mind via computing.

Every type of organization has similar office and administrative processing needs: schools, hospitals, factories, local governmental agencies, farms, research institutions, etc. These processing needs include not merely numerical processing but word processing systems. It is prudent to assume that when scanners capable of reading text are less expensive, the processing of words will exceed the present processing of numerical data. Here will be another plateau in the reshaping of men's minds via computing.

The Reshaping of Institutions

In order to briefly review this important area, the general listing of computer applications mentioned earlier (reference 7) follows.

NUMBER OF COMPUTER APPLICATIONS PER FIELD

(1) Identification	(2) Field of Application	(3) No. of Computer Applications Listed Here
I. Business and Manufacturing in General		
1.	Office	121
2.	Plant and Production	79
	Subtotal	200
II. Business – Specific Fields		
1.	Advertising	22
2.	Automotive Industry	54
3.	Banking	46
4.	Educational and Institutional	101
5.	Farming	44
6.	Finance	41
7.	Government (local, state, federal)	210
8.	Health and Medical Facilities	89
9.	Insurance	44
10.	Labor Unions	11
11.	Law	31
12.	Libraries	12
13.	Magazine and Periodical Publishing	26
14.	Military	12
15.	Oil Industry	83
16.	Police	29
17.	Public Utilities	50
18.	Publishing	11
19.	Religious Organizations	62
20.	Sports and Entertainment	86
21.	Steel Industry	31
22.	Telephone Industry	21
23.	Textile Industry	16
24.	Transportation	174
25.	Miscellaneous	137
	Subtotal	1451
III. Science and Engineering		
1.	Aeronautics and Space Engineering	126
2.	Astronomy	16
3.	Biology	30
4.	Botany	3
5.	Chemical Engineering and Chemistry	36
6.	Civil Engineering	89
7.	Ecology	10
8.	Economics	8
9.	Electrical Engineering	36
10.	Geology	14
11.	Geophysics	7
12.	Hydraulic Engineering	27
13.	Marine Engineering	31
14.	Mathematics	44
15.	Mechanical Engineering	37
16.	Medicine and Physiology	191
17.	Metallurgy	3
18.	Meteorology	25
19.	Military Engineering	30
20.	Naval Engineering	7
21.	Nuclear Engineering	19
22.	Oceanography	23
23.	Photography	15
24.	Physics	34
25.	Psychology	27
26.	Sociology	5
27.	Statistics	27
	Subtotal	913
IV. Humanities		
1.	Anthropology	2
2.	Archeology	17
3.	Art	11
4.	Games of Skill	9
5.	Genealogy	3
6.	Geography	4
7.	History	13
8.	Language	14
9.	Literature	10
10.	Music	19
	Subtotal	106
	TOTAL	3672

Here are a few selected categories for study of institutional applications:

Banking.....46
Educational & Institutional.....101
Finance.....41
Government (Local, State & Federal).....210
Health and Medical Facilities.....89
Law.....31
Libraries.....12
Police.....29
Religious Organizations.....62

These applications affect the people who work in the institutions, the people they serve, and the institutions themselves. Since higher education prepares people for these positions, there is an involvement by undergraduate education. We can see more readily the intimate interdependent relationship of higher education to these institutions within this frame of reference.

The Reshaping of Computer Science

There are vast differences in the curricula of computer science departments, depending upon the nature of the institution. Junior colleges offer more immediately practical offerings than four-year institutions. But state colleges and universities do offer practical job-oriented curricula. The latter broaden the possibilities for study in graduate offerings, often encouraging interdisciplinary studies and advanced applications of the computer. Universities

offering Ph.D's tend to focus on the theoretical, with the intent to prepare four-year graduates for doctoral programs.

All these programs have been reshaped by the needs of industry, governmental institutions, and by the advances in hardware and software. The recent curriculum development of the IEEE Computer Society[9] and the reports of the NSF/ACM group reveal tendencies of integration of hardware, software, and applications. The IEEE is still predominantly a hardware-oriented society, although it is awakening to computer appreciation and social impact curricula[10-11]. ACM planning tends to be more balanced, more systems oriented, with a liberal dash of concern for society.

There appears to be a massage or reshaping of computer science as a discipline, partially because the majority of the programs are in undergraduate institutions where the prime purpose is to prepare people for professional berths in industry. There appears to be more concern for efficiency and job opportunities than concern for using the technology of computers in the service of man, although ACM valiantly carries the flag of social responsibility into the battlefield of technology.

The current reshaping of computer science curricula relates to micro systems, the need for hardware-software graduates, business-oriented systems, and a need to develop technical communications curricula, so that graduates can communicate with users to develop user-oriented systems.

Although specialization still is common in departments, the tendency to encourage interdisciplinary studies is evident. This reshaping of purpose is the result of the massage by industry -- it needs human beings who can design systems, people who can talk to other people who are not programmers. The day of the high priest, who communed intimately with the computer in machine language, appears to be ending.

Computer science is being reshaped by the needs of the outer world. It needs additional reshaping by having faculty who are involved in real-world activities, who know what goes on outside the academic world.

There is still the tendency for academicians to think they know more than the outside world. A collaborative massage or reshaping of business and computer science would enhance the teaching of computer science.

What are the predictions for changes and reshaping of computer science? User-oriented systems will occupy the talents of computer professionals. There will be a greater interface with non-computer people. Computer science will become increasingly planted in the real world, and simultaneously in academia. Its structure will change markedly in the next ten years, due to the needs of people who hire graduates. Hardware will take over the tasks of many present software systems, and a mushrooming of specialized systems for diverse applications will continue. Natural languages will reorder and reshape the character of computer science, altering its scope. Eventually the absorption of computer science by growing numbers of

individuals will reshape this discipline, directing its applied energies into the service of people and their declared needs.

The Reshaping of Higher Education

The Conferences on Computers in Undergraduate Curricula have reshaped the direction of computing in higher education. EDUCOM conferences and regional consortia have altered and enhanced higher education. CONDUIT, with its sharing of tested and documented programs, has served as a model for a center where people can obtain adequate programs that are highly transportable. The ending of cottage industry in computers in education is visible. Seminars and workshops sponsored by the National Science Foundation, innumerable universities and consortia, have enabled a growing number of faculty to become "computer literate". These "converted" faculty have become more facile in computing, and have continued their research with the aid of the computer. They now teach computer applications in their course offerings. Faculty continue to be key persons in reshaping higher education, and thousands of students benefit from enhanced education via faculty involvement in computing.

The appointment of computer instructional coordinators is becoming more common and is no longer considered a luxury. There is still the problem of finding the means (money or release time) to teach faculty about computing.

Although there is a strong trend towards encouraging computer use by growing numbers of faculty, facilities are limited, even sparse in many institutions. Micro systems and the decreasing costs of terminals and mini systems should offer the means of computing for smaller institutions and classroom use.

When facilities are abundant, and faculty are knowledgeable about computing, there will occur a radical reshaping of higher education curricula. Institutions will get what they pay for.

The sharing of software, as represented by EDUCOM, CONDUIT, and various user's groups, can reshape and revitalize computing in higher education. More participation in conferences, consortia, and professional groups will bring this advancement about and afford an economical reshaping of the curricula via computing.

Computers, Instructional Technology, and Learning Centers

The past decade has seen the emphasis of computing for the sake of computing. Computer-assisted instruction on a wholesale scale has been implemented in a limited number of institutions, due to funding. Innumerable conferences have extolled the virtues of computer-assisted instruction. An assessment of CAI was held at the University of California at Irvine in 1975[12]. The brief proceedings of this conference are mandatory thought provoking reading for persons interested in the reality of CAI implementation.

Since the 1975 assessment, the rise of micro systems has alleviated the exclusive concern for "cost-effectiveness" of CAI systems. Micro systems can supply the need for abundant computer-assisted instructional systems. Like the Depression slogan, "a

chicken in every pot", the newest promise is "a micro system in every classroom". This may be a realistic solution for reshaping higher education, and even eduction in general.

But something new has occurred on the CAI horizon. The implementation of computer-managed instruction (CMI) and video-assisted instruction (VAI) has become popular because they appeal to more students than the limited non-visual CAI systems. CAI alone, as an isolated medium, cannot fulfill the multi-sensory needs of human beings. Computer people tend to be so massaged by the machine that they become like lovers, blind to the flaws of the beloved. Computer people have inadvertently placed too much emphasis on speed and the processing of numerical data.

Computer systems without graphic capabilities are not as effective for learning and problem solving[13]. Interactive CAI systems with graphic capabilities and with user-oriented software for specific groups have emerged as the result of need. There has been a reshaping of CAI, CMI, VAI, etc., because of user response to these systems. This enhanced emphasis on multi-media, rather than on one isolated medium, has fostered Learning Resource Centers, Instructional Media Centers, Centers for Information Studies, etc.

Personalized instruction has been wedded to CAI[14]. The need to meet the needs of all kinds of students, to effect the absorption of required specific skills by all ages of students, has massaged higher education. Flexible study periods; self-testing; tutorials; self-paced learning; varied media presentations; learning for pleasure; lifelong learning -- all these are student needs that have finally aroused the concern of higher education. Greater numbers of minority students, older women returning to college, middle-aged students seeking renewal through job changes -- these are a few elements that have reshaped higher education, reshaped computer-related instruction.

The computer has been absorbed into instructionally related environments as a natural part of learning resources. CAI has been reshaped by the people and their needs.

Computer Literacy

At the present time, all universities and colleges offer variations on courses that may be called "computer literacy" or "computer exposure" courses. These range from specialized offerings, accentuating technical material for computer science people ("Introduction to Computers"), to esoteric curricula where art or music afford a pleasant exposure to computer use.

The Association for Computing Machinery has been diligent in assisting institutions to formulate curricula for computer literacy courses. Their new bibliography, available on microfiche, is a generous addition to this curriculum planning.

There is a recognizable gap in computer literacy between generations -- students entering high school, junior colleges, and four-year institutions are exposed to the computer. Older people generally do not use the computer and do not understand it. So far,

there is little provision for this latter group to become "computer literate".

In order to make maximum use of these new systems, small schools need to join regional consortia, to become members of user's groups, to become more knowledgeable, and to secure transportable software for these systems.

Computers and the Humanities

Since this is one of my areas of expertise, I will give slight but special attention to computers and the humanities. The computer is used predominantly for utilitarian purposes. A brief glance back at the numbers of uses in the humanities compared to those in other applications is sufficient to convince anyone of the trend towards utility in computer usage. Of the 2672 quoted computer applications, only 106 represent humanistic uses, roughly 4%.

What is important in humanistic applications is the union of art and science, of the irrational and the rational. The scientific realm is a partial perception. The artistic realm is equally partial. Together the union of art and science form a more complete human perception.

In using computers in the humanities, there is a need to define the limits, the steps to solve a problem (whether that problem is a poem, choreography, sculpture, etc.), and to use logic, reason, and art simultaneously. This enhanced activity has enriched the humanities and reshaped human expressions.

There have been vast changes in humanistic applications of computers during the past ten years. One practical way of assessing these changes is to examine the sprin bibliography issues of Computers and the Humanities magazine, edited by Joseph Raben[15]. Another method of ascertaining this change, this massage via computing, is to examine each program for the International Conferences of Computers and the Humanities, a biennial series of international conferences.

We can observe a new breed of humanist emerging, one who has absorbed computer technology, a person knowledgeable in specific humanistic areas, and equally familiar with computer systems.[16] Transdisciplinary research and creativity, the computer and dissimilar disciplines, are now typical in humanistic research. The following is from the Second International Conference of Computers and the Humanities. Humanists who use the computer have progressed beyond "amateur" computer capabilities, to become literally computer professionals. In other words, they have absorbed computer technology, and from this usage, there has been evident reshaping of humanistic scholars' minds.

INTERNATIONAL CONFERENCE OF COMPUTERS AND THE HUMANITIES
CONFERENCE SCHEDULE
University of Southern California, Los Angeles, Calif.
April 3-6, 1975

Computer-Assisted Instruction 1

Tej Bhatia (Univ. of Illinois)
NEW DIRECTIONS AND ISSUES IN CAI

Ellen W. Mold (Stanford University)
FEAR AND TREMBLING: THE HUMANIST APPROACHES THE COMPUTER

George M. O'Brien (Univ. of Minnesota, Duluth)
SIREN SONGS AND A SKEPTIC

Data Bases 1

Theodore F. Brunner (Univ. of Ca., Irvine)
THE THESAURUS LINGUAE GRAECAE PROJECT: May, 1975

Robert Benson (Univ. of Calif., LA)
A PROPOSED COMPUTER THESAURUS OF MEDIEVAL LATIN

Joseph Donohue (Univ. of Mass.)
THE LONDON STAGE 1800-1900

Art 1

Ruth Leavitt (Univ. of Minnesota)
PLASTIC DEFORMATION OF DESIGN

Hiroshi Kawano
MARKOV PROCESS THEORY OF PICTURES

Leonard Meyers (San Francisco State Univ.)
COMPUTER-ANIMATED FILM AS VISIONARY ART

Music 1

Garrry E. Wittlich (Ind. Univ., Bloomington)
DIRECT KEYBOARD INPUT OF MUSICAL SCORES

Jerome S. Wenker (Sperry-Univac Corp.)
ON THE ANALYSIS OF MUSICAL ANALYSIS

Raymond F. Erickson (Queens College, CUNY)
THE DARMS PROJECT

Stylistics 1

Colin E. Martindale (Univ. of Maine)
THE NIGHT JOURNEY: PATTERNS OF REGRESSIVE IMAGERY IN LITERARY JOURNEYS TO HELL AND THEIR ANALOGUES

Daniel L. Greenblatt (Univ. of Missouri)
VARIABLE RULES AND LITERARY STYLE

David H. Chisholm (Univ. of Arizona)
PHONOLOGICAL PATTERNING IN GERMAN VERSE

Stylistics 2

James Joyce (Univ. of Calif., Berkeley)
COMPUTATIONAL MODEL OF STANZAIC PATTERNS IN ENGLISH

Jay A. Leavitt (Univ. of Minn.) and David Hacker (Univ. of Iowa)
ON THE MEASUREMENT OF ALLITERATION IN POETRY

Leslie A. Olsen (Univ. of Michigan, Ann Arbor)
SOME SIMILARITIES BETWEEN WORDSWORTH'S AND MILTON'S BLANK VERSE PROSODIES

Computer-Assisted Instruction 2

Peter Zoller (Wichita State University)
A CAI APPROACH TO BLACK ENGLISH

Fred T. Hofstetter (Univ. of Delaware)
DEVELOPMENT OF A CENTER FOR COMPUTATIONAL MUSICOLOGY AT THE UNIV. OF DELAWARE

Robert L. Oakman (Univ. of South Carolina)
A VIDEOTAPE COURSE FOR COMPUTER EDUCATION IN THE HUMANITIES

Art 2

Luraine R. G. Tansey (San Jose City Coll.)
PRE-COLUMBIAN ART: SITES AND CHRONOLOGY COMPUTERIZED

Eleanor Guralnick (Univ. of Chicago)
THE PROPORTIONS OF ARCHAIC GREEK SCULPTURED FIGURES: A COMPUTER STUDY

There have been many changes in the types of research in the humanities. There is still evident a great preoccupation with the past, with tradition. An analysis of computer activity over a period of the last ten years, attained by a study of bibliography material from Computers and the Humanities, reveals a marked tendency of scholars to focus on art and ideas of the past, of the classical, of the securely ordained "great ideas", with a limited tendency to think of today and of what humanism means in a cybernetic society.

There is a tendency, as in computer conferences, to use the scientific method, to report theories, experiments, and precise outcomes, without synthesizing the parts of the whole. There is a need to restate what the humanities mean, only this time in relation to a cybernetic environment. I have written papers on this subject, but have felt that philosophical discussions too often fell on deaf ears[17]. The Europeans appear to be more interested in the ideas behind computing and the humanities, and their conferences are more intriguing intellectually and philosophically[18].

Actually, a connection between the visual arts and computers is very highly developed. Twelve years of refinement have brought about marked changes in the development of user-oriented systems[19] and in methods of final presentation[20]. The visual arts have been taken back to art, and the computer technology has been absorbed by practitioners, nurturing a humanist pattern with interdisciplinary capacities.

Humanists are now forced to ask new, more significant questions, and are proceeding beyond myopic concerns with counting words and cataloging scenes. The problems with computers and the humanities have become more significant and more contemporary. Humanists have exquisitely absorbed computer systems and languages, to become "artists" in their own right. Perhaps computer scientists could emulate this absorption of the computer, to proceed beyond excessive reverence for the machine, to consider its humane uses, and to ask "Why am I computing?"[21]

Interdisciplinary Absorption of Computer Technology

The previous discussion leads naturally to a consideration of the reshaping of diverse disciplines via computing. In studying the table of applications, the concerns of business applications, one proceeds to an examination of the scientific fields. The relatedness of the academic world to the scientific applications is immediately apparent, for in order to exercise computer applications in these areas, previous professional preparation is required. There is a need for computer absorption within these specified applications and fields. A few days ago, I asked a visiting speaker, Melvin L. Prueitt, a physicist from the Los Alamos Scientific Laboratory, to discuss computer applications in his area[22-23]. Dr. Prueitt casually stated "I regard the computer as a tool. I just use it. It's like a typewriter."

But too many institutions have not caught up with the requirements for computing in their disciplines. There is still a gap between real world applications in diverse fields and what is taught in academia.

The faculty may well ask, what are requirements in my field? How can I know what I should know?

A definitive technical report, "Recommendations for Computers in Higher Education", will be released in the early autumn of 1978. This study reviews every department in a university, every class, and makes recommendations for computers in these courses. I began coordinating this study with the help of juniors and seniors in a class called "The Societal Effect of Computers".

As people from diverse disciplines use the computer, more information is processed, and more is known than before. There is a marked tendency not merely to collect information (as in manual processes), but to consider the signficance of the problems under study. It is no longer adequate merely to collect and process data, and the computer has reshaped many disciplines in this significant manner.

The methods of thought referred to earlier, the algorithmic pattern and the "what if" pattern, are significantly changing disciplines. Further, this combination of computer science and new disciplines is nurturing other transdiscipline studies of dissimilar natures. Examples are traditional and non-traditional medicine; business and social responsibility; agriculture, computer science, and business; preventive medicine and psychology, etc.

Thus the minds of men have been reshaped by absorption of computer technology and by a revitalization of new approaches to problem solving, to the formulation of innovative problems. Man is moving towards a wholeness of approach that departs drastically from the compartmentalized disciplines characteristic before computer usage. The computer is a catalyst for change.

Natural Languages and the Reshaping of Computers

I would like very briefly to mention a reshaping of the mind and a reshaping of computers that is inevitable: the breakthrough of programming in natural languages. The literature is lean in these areas, revealing complex difficulties in using natural language input for computer processing. But this area of research is like an iceberg, with the surface pattern barely visible above the water. In reality, the entire pattern is submerged, but definitely there. In half a dozen laboratories around this country, progress in computer programming in natural languages has been extensive. However, these confidential proprietary results are not publishable at this time, because publication would alert competitive laboratories and researchers, and allow competitors to utilize the research of the writers. This is an interesting example of the silence needed in industrial research. Here is a brief example: recently a graduate student wanted to write a master's thesis on comparisons between computer applications of the oil industry in the United States and Iran, and to predict potential new areas of development for the latter country. He was unable to find material on the subject, even with the help of a professional librarian who was experienced in automatic researches. As the student's committee chairman, I took him to the library, to see for myself. There was nothing! I began phoning around the country, talking to various oil firms, and the story was

the same: computer applications in the oil industry are vast and highly sophisticated, but nothing is published because of its proprietary nature. This is an example of how deluding a review of the literature may be. The state of the literature is often not the state of the art.

There are two brief items on natural language programming, both very partial: Computers and People, September, 1976, "Programming in Natural Languages, A Brief Report"[24]; and an advertisement, "A Black Box Called Djinni" by Berkeley Enterprises, Incorporated[25]. Bear in mind that these materials are very limited in information, because of proprietary interests. But the iceberg is quite large and very important.

Since the reader is probably a computer science person, I would ask, "What do you think the effect of programming in natural languages will be on this discipline?" We cannot accurately predict the outcomes, but we do know that this will be a revolutionary reshaping of computer science. Many techniques will be obsolete, and new capacities will be required of computer people beyond the present skills.

The reshaping of computer science for the user may mean freedom from programming, freedom to massage and reshape computer applications for man's myriad purposes. "Myriad" means specifically ten thousand. Since I have prudently estimated that the number of current applications is 5000, to double this amount is a very modest estimate. When this new capacity to process information via natural language takes place, the massage and reshaping of man's mind and computers may have come full circle. When we take natural languages and inexpensive micro systems back to the people, back to the users, the reshaping of disciplines, techniques, education, business, institutions, etc. will be significant.

Computer Science and the High Schools

Mini computer systems have been purchased for administrative purposes in a growing number of high schools, but usually a portion of computer time is allocated to instructional computing. During the past six or seven years, introduction to computers was designated for the "mathematically gifted". Experiments in high schools have altered this view, and computers are now utilized for business, mathematics, graphics, etc[26].

Due to the availability of systems, and the interests of high school students, secondary school curricula are being reshaped via computers. It is now feasible to implement computer-related instruction on a much wider scale[27]. For the first time, even "poor" schools can afford the luxury of computers in the classroom via microcomputers.

Joyce Statz and Leland Miller of Bowling Green State University have reviewed this trend of computers in high schools within a generous study titled "Proposal for Certification in Computer Science"[28]. They quote Charles Darby and his 1972 study as follows:

> It can be projected that within the next decade every secondary school in the country will have access to a computer system for some type of administrative and/or instructional application.[29]

The need for professional computer science teachers in the high schools is a reality now.

In the United States, computer science teaching credential programs for the secondary schools have been developed and implemented in Minnesota, Ohio, Oregon, Texas, and Wisconsin[30]. A new credential program is planned for California; it differs from the other programs in that it is the only five-year plan, because of credential requirements defined by the Ryan Act, passed in 1970.

Here are a few specific details of each program, given very briefly:

CALIFORNIA - California State University, Chico, has begun planning for a single subject credential in computer science. It will offer a concurrent master's degree in computer science and a high school/junior college teaching credential in computer science. All other programs are related to business or mathematics. An unpublished paper, "Some Questions: Computer Science Secondary Teaching Credential Curricula"[31] reviews in depth each program, the need for standards, etc. The author is the originator of this curriculum planning.

MINNESOTA - The computer science teaching degree program at Mankato State University is designed for people who desire to utilize, teach, or coordinate the computer science curriculum of a junior or senior high school. Each major must complete a related academic core in mathematics or business education in conjunction with this new degree program. The program is based upon a computer science major.

Here is an important (updated) statement from the Minnesota program literature:

> The Association for Educational Data Systems states there will exist over the United States a minimum need of 50,000 more teachers of computer science within five years.[32]

OHIO - Bowling Green State University is the only Ohio institution offering a degree in Computer Science in Education. Students have obtained their degrees in the College of Arts and Science. The four-year Ohio program differs from the Minnesota program in that it has two options, a major or a minor in computer science. This program offers a strong background in computer science as the basis for its credential preparation.

OREGON - The Oregon Council for Computer Education commissioned a study on the question of a computer teacher norm. David Moursund, University of Oregon, has been very active in working with secondary school mathematics teachers and conducts summer workshops for them. Mr. Moursund has written several papers regarding computing needs in the high schools, and has well developed statistics regarding these needs [33-34].

Under Dr. Moursund's leadership, Oregon has begun a program for teachers of grades 1-12 at the University of Oregon. Planning is still in progress, and specific details of credential requirements are not yet available from this source.

TEXAS - Texas uses computers in the secondary schools, and these applications are now being studied by the Texas Education Agency. This agency has recommended the formation of a full-blown teaching credential, and the planning at Southwest Texas University's Department of Mathematics is for a second teaching field, computer science and mathematics/business credentials. The curriculum planning is for a secondary computer science provision program. Like the other programs, it is a four-year program.

WISCONSIN - The Computer Science Department at the University of Wisconsin at LaCrosse offers a credential program that offers a major and minor in this field. Graduates of this program teach computer science and mathematics. This is a four-year program, based upon professional preparation in computer science.

Summary

With the introduction of credential programs in computer science for secondary schools, the content and form of many subjects will be altered.

There are, at the present time, no uniform accepted standards for computer science teaching credential programs at the secondary level. We may anticipate formation of standards by ACM and also by special interest groups in this area, in an attempt to establish specific minimum standards and guidelines for future development of these programs.

In essence, computing is leaving higher education for the lower levels of education. The work of David Moursund and others, cited in the preceding section, has established a stated need for computers at every level of education.

The drawbacks have been the cost of hardware, lack of software, lack of trained teachers to introduce computer science to students and to other teachers. With the availability of micro systems, transportable software, and institutional programs to teach graduates how to teach computer science, this is changing radically.

In addition, many students will not go on to higher education. These people have a need for computer literacy in order to survive in a technological society. Therefore, computer exposure should be at pre-college levels.

Present reshaping of curricula in the elementary and secondary schools is subtly altering these levels of education. What happens at these levels will profoundly alter what happens at higher levels of education.

Anticipated Changes in the Undergraduate Curricula

The thrust towards procurement of hardware for computing at lower levels of education will move computing into these areas. The availability of teachers for these levels of pre-college education will alter and reshape what is taught in the undergraduate curricula. Students will enter higher education with computer literacy courses and programming experience.

In-service courses will expose teachers to computing at every level. In colleges and universities, teachers will be expected to use computers and to teach computer-related courses.

It will be far more common for students and teachers to cross disciplines, to use computers plus secondary and tertiary disciplines in their work. The closed discipline/department will become obsolete.

Instructional media centers and other centers of learning will offer CAI, CMI, and VAI courses using programming languages and personalized instructions, with alternatives of presentation and flexible times for study. Not all students want to use computers; many prefer video instruction or other methods not related to computers. Self-testing will replace present teacher-testing in the classrooms.

The majority of schools have not met the needs of diverse groups of students that are admitted to universities and colleges under the open admissions programs. There is visibly a thrust to hold institutions <u>accountable</u> for imparting specific skills and well developed learning capacities.

Computers and Education of the Future

This material on anticipated changes in the curricula has been paraphrased from the earlier unpublished paper on credential programs by the author.

From mere usage, the computer evolution will revitalize and reshape education of all levels. Transportability will be required in curriculum planning and program development. The mystery of the black box will be a thing of the past. Computers will become as natural as electricity in the classroom. Conferences on computers in the undergraduate curriculum may become extinct in their present form. In order to survive, they will have to focus on transportability and shared program development. Computers in the humanities conferences will be altered in that utilization of the computer in varied humanistic disciplines will be an accomplished fact. Esoteric developments in the humanities will rise, like futuristic sculpture, to be aired in these international gatherings. EDUCOM should become more important with the development of regional centers for gathering of programs and dissemination of systems to member institutions.

Standards for languages will be developed, and less cottage industry will be visible. Faculty will be discouraged from developing idiosyncratic systems that re-invent the wheel. Computers in education will eventually mature to the place where structure, form, and principles of technology and learning will become common and

accepted. This will afford a basis for a foundation of knowledge and techniques, so that new structures will proceed beyond present systems. An example is structured programming, which will become mandatory in the profession of computer science. We have finally matured to the place of forming principles of programming. This will also be true of computers and learning.

However, the isolated role of the computer will be reshaped by implementation with other technologies. The computer and other technologies will be firmly placed in the classroom, along with other visual aids and learning systems. Perhaps we have placed computers above other forms of learning. The need to take computers back as a natural part of the educational environment is obvious.

The final massage will be absorption of computing and absorption of technological systems within learning so that we do not write and hold conference about it.

The Reshaping of Man's Mind - Summary

In November, 1961, I attended a Teachers' Convention in Milwaukee, Wisconsin. The guest speaker was Werner Von Braun. I will paraphrase what he said. It made an indelible impression on my mind: "The future is so vast, that all we can do is to teach people to think and to problem solve. If we accomplish this, people will survive."

In using the computer, we have made significant advances in teaching people to think and to problem solve. These advances would not have been possible without the computer. The specific advances we have made in thinking and problem-solving were made possible by the interaction of man's mind and the computer.

If we can continue our present absorption rate of technology and computers, there is added hope for man to solve the significant problems of the earth.

The reshaping of man's minds via computing represents an enhancement, a revitalization, an extension of men's thinking and inventing capacities.

We are presently standing on the shore of a vast new continent, untrod by men before. We are presently colonists of a new earth-world, made possible by the reshaping of men's minds via computing.

References

1. McLuhan, Marshall and Fiore, Quentin, The Medium Is the Message (New York: Bantam Books, 1967).

2. McLuhan, Marshall. The Gutenberg Galaxy (New York: New American Library, 1964).

3. Meadows, Dennis L. The Limits to Growth (New York: New American Library, 1972).

4. Negoponte, Nichols, "On Being Creative with Computer Aided Design," Computer Graphics and Art, Volume 3, No. 1, February, 1978; also Proceedings IFIP Congress 77, Toronto, August 8-12, 1977, IFIP North-Holland Publishing Company, edited by B. Gilchrist.

5. Berkeley, Edmund C., Giant Brains (New York: Science Editions, Inc., 1961).

6. Berkeley, Edmund C. and Hertlein, Grace, "Computers", Social Consequences of Engineering, edited by H. Kardestuncer (New York: Harper and Row, 1978).

7. Berkeley Enterprises, Inc., Newtonville, Mass., The Computer Directory and Buyer's Guide (Boston Mass.: 1974)

8. Berkeley, Edmund C., and Hertlein, Grace, op. cit.

9. IEEE Computer Society, "Proceedings of the Computer Science and Engineering Curricula Workshop," June 6-7, 1977, Williamsburg, Virginia. IEEE Service Center, Piscataway, New Jersey.

10. Hertlein, Grace C., "Engineers, Computers, Social Responsibility," Proceedings, IEEE Curricula Workshop, June 6-7, 1977, Williamsburg, Virginia.

11. Austing, R.H.; Cotterman, W.W.; and Engel, G.L., "Literature Resources and Impact on Society and Computer Literacy," Proceedings, IEEE Curricula Workshop, June 6-7, 1977, Williamsburg, Virginia.

12. Feldman, Julian and Mosmann, Charles (Editors), Instructional Computing in the University - The Second Decade (San Francisco: San Francisco Press, Inc., 1977)

13. Bork, Alfred. "Learning through Graphics," Computer Graphics and Art, Volume 1, No. 1, February, 1976.

14. Pessel, D., "Introduction to Computing Self-Paced: A Novel Approach Which May Be Fun But Does It Work and Is It Worth It?" Proceedings, IEEE, Computer Science and Engineering Curricula Workshop, June 6-7, 1977, Williamsburg, Virginia.

15. Computers and the Humanities, Edited by Joseph Raben, Queens College, Flushing, New York.

16. Lusignan, Serge & North, John (Editors), Computing - The Humanities (Waterloo, Canada: University of Waterloo Prov. 1977).

17. Hertlein, Grace C., "Renaissance/Reformation: The Humanities Within a Technology-Oriented Society," Proceedings, ICCH/1, International Conference on Computers in the Humanities, University of Minnesota, Minneapolis, Minnesota, July 20-22, 1973.

18. _____, Art/Science: "The Union of the Rational and Irrational," Proceedings, International Conference on Computer Art -- Tendencije 5, Zagreb, Yugoslavia, June 1-July 1, 1973.

19. _____, "Design Techniques and Art Materials in Computer Art - Part I," Computer Graphics and Art, Volume 2, No. 2, August, 1977.

20. _____, "NCC Art Review, Directions and Questions: The Role of the Computer in Computer Art," Computers and People, August, 1976, Volume 25, No. 8.

21. Klein, Sheldon, "Automatic Novel Writing: A Status Report," ICCH/1, University of Minnesota, Minneapolis, Minnesota, June, 1973 - Technical Report #186, Computer Sciences Department, University of Wisconsin, Madison.

22. Prueitt, Melvin L., "From Data to Dimension: Computer Drawn Graphics for Science," Functional Photography, January, 1976.

23. _____, Computer Graphics, (New York: Dover, 1975).

24. Berkeley, Edmund C., "Programming in Natural Languages, A Brief Report," Computers and People, Volume 25, No. 9, September, 1976.

25. Advertisement, Berkeley Enterprises, Incorporated, "A Black Box Called Djinni," Computers and People, Volume 26, No. 10, October, 1977.

26. Chu, Kai, "Computer Graphics and High School Education," Computer Graphics and Art, Volume 1, No. 3, August, 1976.

27. Moursund, David, "Computer Science Education of Secondary School Mathematics Teachers, Oregon Computing Teacher, Volume 5, No. 1, October, 1977.

28. Statz, Joyce and Miller, Leland, "A Proposal for Certification of Secondary School Computer Science Teachers in the State of Ohio," Undated document, 17 pages, approved June 14, 1976.

29. Darby, Charles A., Jr.; Korotkin, Arthur L; and Romasko, Tania, The Computer in Secondary Schools, A Survey of Its Instructional and Administrative Use (New York: Praeger, 1972).

30. Hertlein, Grace C., Unpublished Paper, "Some Questions: Computer Science Secondary Teaching Credential Curricula," Spring, 1978.

31. Ibid.

32. Brochure, "Computer Science Education" - Bachelor of Science Degree in Computer Teaching, Department of Computer Science, Mankato State College, Mankato, Minnesota (Undated).

33. Moursund, David, "Opinions on Computer Science Courses for Precollege Level Teachers," ACM/SIGSCE, Detroit, Michigan, February, 1978.

34. _____, "Computers in the Elementary School: A Course for Teachers," OCCE Special Report, January, 1975, University of Oregon.

COMPUTERS, EDUCATION, AND THE FUTURE OF EDUCATIONAL INSTITUTIONS

by

Alfred Bork
Department of Physics
University of California
Irvine, California 92717
(714) 833-6911

Introduction

The digital computer is one of the great toys of our age. Like all toys it will have great effects on our society. We can see it already playing a larger and larger role in all areas. My present concern is particularly with education and, to a larger extent, higher education. But many of the conclusions that I will present are also applicable to all levels of education going from early childhood through adult education.

The computer will profoundly affect not only the courses in colleges and universities, but even the structure of these institutions. This effect will be due not only to the computer; other very powerful factors will be working to change education in the 1980s. Computers will play a role in creating some of the problems and in providing modes of solution to these critical problems.

American institutions of higher education in 1995 will not look like those we see today. Our institutions will be different in ways that will be startling, particularly considering the almost unchanging nature of these institutions in the last century.

Not all of our current institutions will survive these changes. Some will resist the forces suggesting change, and many of these that resist will die. Other institutions will survive only in crippled forms. The vigorous institutions of the future will be those that

understand the problems and have the leadership to carry out effective solutions.

What is the time scale for this change? In a study done five years ago by the Carnegie Commission on Higher Education, The Fourth Revolution - Instructional Technology in Higher Education, it was predicted that the period from 1990 to 2000 would be a period of very rapid introduction of computers in the learning process in colleges and universities. That report argues that after the year 2000, computers will be very heavily used as a standard part of the curriculum. This projection is now too conservative. The changes appear to be coming somewhat faster, particularly the development of microcomputer technology. Probably we can take five years off each of the Carnegie figures.

The hardware directions for these changes are now clear. While even in as short a time as five years ago, it appeared that timesharing systems connected to central computers would be the principal student delivery system, the developments in the past few years make it likely that stand-alone computer systems, occasionally interconnecting for special purposes, will be the dominant student delivery hardware. Developmental work will often still take place in timesharing environments. Other speakers at this meeting have emphasized the importance of personal computers, so I need not dwell further on its critical role in future education.

I have just made some startling statements. The remainder of this talk will be devoted to elucidating and supporting those statements.

Why the Computer?

Academicians have been hearing about revolutions in education for a long time. In particular, educational technology has often provided us with predictions of such revolutions. Thus, video technology, permitting campuses to be connected with video systems, was proposed as a revolutionary force in education. There are some campuses of this kind already. Recently the Sloan Foundation provided money to the Massachusetts Institute of Technology and Dartmouth College to construct TV networks; many other institutions, such as my own, the University of California at Irvine, have provided such connections partially through existing cable TV systems in the area.

Yet the changes that have occurred because of past educational technology would hardly be described as revolutions. The colleges and universities have hardly changed at all. So announcements of new revolutions, such as the one I am proclaiming, are often greeted with considerable skepticism. Universities are extremely conservative organizations, slow to change. In many cases, the technological innovations have simply rolled off the back of the oily university.

So why should the computer, the newest push of technology on the educational scene, be different from some of these earlier developments? Two fundamental reasons are important.

The first of these is that the computer is a very effective teaching and learning device, allowing effective learning approaches which are not present in many institutions, particularly large

institutions with many students. One can see this effectiveness through examples. I will show you an example of a course at the University of California at Irvine, where the computer aids in many ways by providing a better presentation than would be possible without the computer. I use this illustration not because I regard this course as the ultimate course -- it will be regarded in the future as a bumbling early attempt -- but because it shows the possibilities inherent in this new and exciting learning medium.

The second reason is a more mundane one, but one that will nevertheless become extremely important in the future. Of all the components which go into the educational process, only the computer is coming down rapidly in cost. This decrease is expected to continue for a very long time, making the computer eventually the most cost effective educational mode.

In what follows, I will treat each of these topics in more detail.

A Course with the Computer

The course I will describe is Physics 3A at Irvine, as taught in the fall quarter of 1976. It is a 300-student introductory physics course intended primarily for biology majors. This course was offered only once. The next year we offered a similar course for science and engineering majors, but with some details changed from the course I will show you.

The course attempted, through the use of the computer and other learning modes, to accomplish something that is seldom allowed in American courses, to allow student choice over content. Two major "tracks" of material were available for students to study, each containing eight individual modules. These two tracks were quite different. One was a conventional course, following a standard textbook for beginning physics. The other made extensive use of computer programming as an intellectual aid to learning physics, providing the powerful resources of the computer to students. Various crossovers were allowed between the two tracks, so that the student had effectively a choice of six different routes between various components of the course.

In addition to choice of content, the students also had choice of timing, the rate at which they moved through the material. Thus, the course was self-paced. Students were urged to pick a definite pacing strategy as a conscious choice.

The course was also a mastery-based course, a variant of a procedure for teaching courses often called PSI (Personalized System of Instruction) or Keller Plan. Modules are specified, each with carefully stated objectives. Each module must be "mastered" before the student can go on to the next module. Learning is demonstrated by taking a test associated with the module. In our course, as with other courses of this type, the student must perform essentially perfectly on the test to go on to the next module. If students do not, then further study is required, and another form of the test given.

All students in the course had computer accounts. During the ten weeks of the quarter, the average student used about two and one-half hours of time at a computer display per week. Hence, the total time involved with all the 300 students in the class was about 7500 hours in the ten weeks.

The computer was used in a wide variety of modes within the course. First, it was used as a communication device, with electronic mail often going between the students in the class and me. I could broadcast a message to each student in the class to inform them immediately of problems associated with the course, or I could contact students who needed individual attention. Students could send me messages about problems that arose. Typically I would "answer" my computer mail once a day, usually in the evening from home. Computer mail was not intended as the sole communication mode. I was in my office for many hours during the week talking to students individually about problems associated with learning the material. But the computer mail provided an important additional way to communicate.

The second use of the computer within the course has already been mentioned, the use of the student's own programming as an aid to learning physics in one of the available tracks. Thus, not all students used this mode. The programming (in this case the language chosen was APL) was an integral part of the unit. We did not assume that students knew APL before the course; one of the eight units was spent in learning APL. Learning APL was efficient, involving interaction with APL and observation of its behavior. APL was the conscious choice over a great variety of available languages. One important factor, but not the only factor, in this choice was the fact that we had developed good efficient graphic capabilities within APL.

The use of programming within courses provides students with a very important and powerful intellectual resource, enabling a new approach to the content of the material. Thus, the computing track looked very different, allowing the student to move much more quickly to a view of the basic laws of mechanics (Newton's $\vec{F}=m\vec{a}$) as a differential equation, rather than seeing the law as the traditional algebraic statement.

The third role of the computer in the course was as a tutorial device, helping students to learn. A series of tutorial programs was prepared by my colleagues and myself. This type of learning allows individualization of learning; student responses can be carefully analyzed and the program can go off in different directions with alternate student responses to questions posed by the computer. Remedial aid can be provided where needed. Furthermore, and perhaps even more importantly, a tutorial program on the computer is an active learning mode, as compared to passive learning often seen in large lecture courses or with students reading books. The student must always play a dynamic role in the learning process in our computer dialogs. It is a more personalized and humanistic approach to learning than is possible in a 500-student lecture, much more tailored to the needs, pace, and possibly even interest of each student.

The computer allows each student to have a unique experience. Kenneth Eble, in The Perfect Education, comments "Education is personal. At best it is intensely personal." Eble was arguing that the critical component in education can be the real involvement that

the student has with the teacher. But the computer can also be a help to moving toward more personal education.

A fourth role of the computer in the course was its use as an aid to building physical intuition. One difficulty with many university courses is that the material has a high level of abstraction; so the experiences that the student brings into the course do not provide a rich learning background. But the computer can simulate physical systems, and with proper student interfaces, can make it easy for the student to gain additional experience. In this course, the experience we wanted was with the laws of motion. The students could observe bodies moving under the action of many force laws and could study the behavior in space other than ordinary configuration space. We found it necessary, in order to make these programs useable by the bulk of the students, to provide auxiliary materials. The programs by themselves were inadequate. The additional information, in the form of workbooks, told the students what cases to look at and requested that the student write descriptions of what was seen. These materials were criticized by our evaluators, and we are currently in the process of rewriting them. The goal of providing greater intuition for students is important. This goal can be assisted by the computer with the proper associated learning materials.

A fifth, and very important, use of the computer in the course was giving the tests associated with each of the modules. As indicated earlier, students take tests to show they know the material in a PSI course. Alternate forms of these tests must be provided, since a student may typically take a test several times before demonstrating mastery of the material in that unit. We gave all the unit tests directly online at graphic computer displays; there were twenty-seven tests for the two tracks of the course. Tests were individually written and allowed a wide range of student responses. During the ten weeks of the course, we gave over 10000 online tests!

Although I am describing these programs as tests, it would be more accurate to describe them as an intimate blend of testing and learning. They are almost a <u>new</u> educational form, not available without computers. Not only is the student told immediately when right, but also learns immediately when mistakes are made. Often the student is given detailed help, specialized to the particular mistake that the student made. These extensive help sequences are built into the tests in a variety of modes. Although the student must complete a test in forty minutes in one sitting, the help sequences (if accessed) do not count toward this time. A student who is clearly in trouble early in the test will be given advice as to what further study is necessary, and will typically not be allowed to complete the test.

As the online tests were a critical feature, and because such tests are relatively new, I present at this point a sequence of representations of the information on the computer screen. They are taken from a quiz called VECTORS. Unfortunately, they do not show the dynamic aspect of computer dialogs. (Only a part of the exam is shown.)

If you encounter any problems
or have any gripes
keep them in mind!

We'll ask you about them
at the end of the session.

End each answer
by gently striking return.

Try it now!

right	0
wrong	0

Each page will have a scoreboard
in the upper right corner.

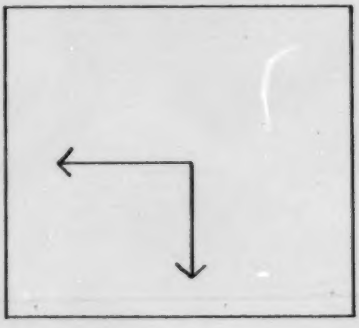
The scoreboard indicates your progress.

You will receive a unique set of questions.

Please type RETURN to get the first question . . .

PROBLEM 1

Consider two vectors --



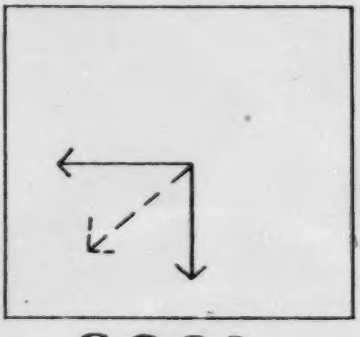
right	0
wrong	0

Point to the tip
of the resultant vector,
using the white thumbwheels
to control the crosshairs.

Press return
when the crosshairs
are
where you want them.

PROBLEM 1

Consider two vectors --



right	0
wrong	0

Point to the tip
of the resultant vector,
using the white thumbwheels
to control the crosshairs.

Press return
when the crosshairs
are
where you want them.

GOOD

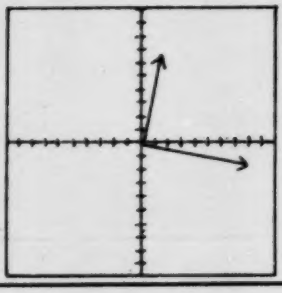
**Press RETURN for problem 2...

PROBLEM 2

right	1
wrong	0

Here are two more vectors --

Again indicate the tip of the resultant.

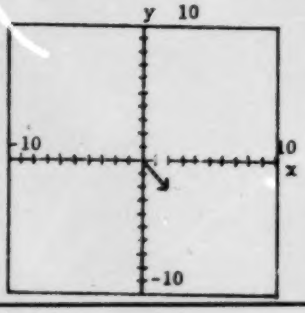


PROBLEM 3

right	2
wrong	0

Your next task is to resolve a vector into components. Here's the vector A.

And the coordinate system.



Give me the x and y components

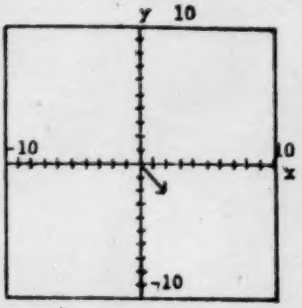
A_x = A_y =

PROBLEM 3

right	2
wrong	0

Your next task is to resolve a vector into components. Here's the vector A.

And the coordinate system.



Give me the x and y components

$A_x = 2$ $A_y = 2.5$

Again!

$A_x =$ $A_y =$

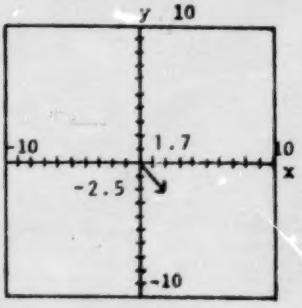
Your career in physics will suffer if you don't pay more attention to signs.

PROBLEM 3

right	2
wrong	0

Your next task is to resolve a vector into components. Here's the vector A.

And the coordinate system.



Give me the x and y components

$A_x = 2$ $A_y = 2.5$

Again!

$A_x = 2$ $A_y = -2.5$

GOOD

RETURN ! ! !

Your career in physics will suffer if you don't pay more attention to signs.

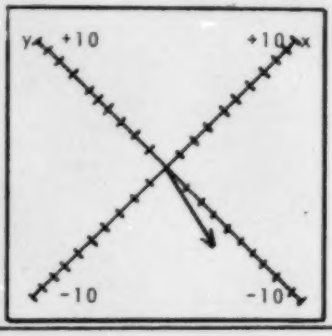
test9

PROBLEM 4

right	3
wrong	0

Here's the same vector

but another coordinate system.



Enter the components . . .

A_x = A_y =

All of the twenty-seven tests were written the summer before the course was given. I emphasize this to show that with an effective system for authoring and producing such computer-based learning material, we can efficiently develop a large body of materials in a relatively brief period of time with finite resources.

The final use of the computer in the course was a full course management system. The results of the online tests were recorded directly into the student data base, and so the student records were available in a machine readable form. Programs were developed to provide students access to their records and to provide information to the instructor. Thus, it was possible to show data on each test in graphic form, and to pick out tests which were too difficult. It was also possible to identify several types of student learning problems, and then to contact students individually by computer or regular mail. A variant, used in the present course, was a lockout on how many times a student could take a particular quiz. During the current year, if the student fails a quiz four times, the computer management system does not let that student take that quiz again until the student has come in to talk with me. Thus, I see individually every student in trouble.

I would not want to claim that these are all the ways the computer could be used in a course. Some of these modes are related to the subject area to some extent. Nevertheless, I present them as a whole to make the case for the computer as a powerful and effective learning aid, and one that can make a qualitative difference in students' ability to learn.

A formative evaluation of the course was carried out by Michael Scriven and Jill Larkin, University of California at Berkeley. Their report was very helpful to us in improving the course material.

The Cost of Computers

The second factor mentioned earlier as being extremely important in arguing that computers will profoundly affect education is the rapidly decreasing cost. Perhaps this can best be illustrated by beginning with the cost of our present mode of doing things. The course that I just described runs on a general purpose machine, a timesharing Honeywell Sigma 7. Like most universities, we have a complex pricing algorithm, an attempt to account for the cost of running the machine. The cost per student with our current system is approximately $75 per quarter. But this hardware is, in design, about fifteen years old. What would the situation be if we ran all the materials on an adequate contemporary system, such as the Terak 8510A, a stand-alone system based on the DEC LSI-11? The system costs (with discount to members of EDUCOM) $5,500. The pricing on this system is such that we could be running our current course for hardware charges of approximately $25 per quarter per student.

This considerable reduction in cost is nothing unusual in the computer field. It is argued that costs for equivalent computer power go down by a factor of ten every seven years. The stand-alone systems particularly benefit, because they are best able to exploit the new technologies, primarily large-scale integration. This technology has put more and more components in less and less room for less and less cost. But remember that my example is based on an extensive use of the computer within the class, and the costs I am quoting are for hardware, including purchase and maintenance, and for general support of underlying software. They do <u>not</u> include curriculum development costs.

If we project these cost estimates to 1995, we can expect very elaborate computer systems fully adequate for computer-based education at less than $1,000, with a cost per student of less than $5.

We cannot claim such dramatic declines in the production of courseware. But nevertheless, we are becoming more skilled in courseware production, in developing efficient production systems, and in studying the various components needed to perform the task. The cost factors are very much dependent on scale of use; as we move to more and more widespread use, the cost of developing materials will be effectively less for the individual user. An important factor will be the home market and other public computer uses. As more computers are available in these locations, materials will be developed for these markets. These activities will further develop our skills in developing materials, and so will lead to practical courseware.

All these factors point to computer-assisted learning modules becoming more and more competitive with other kinds of modules. So the computer materials will eventually furnish not only a desirable way to learn, but also the least expensive way to learn.

Higher Education Today

The dramatic rise of computer technology is not the only important factor in the current educational situation. Indeed, seldom has there been a period where universities are more beleaguered by external circumstances than they are today.

One of the factors associated with universities that has received the most publicity in recent years is the demographic situation. Fewer students will be available. What is so impressive about this information is that no speculation is involved. The students that are being talked about are already alive. There are simply not as many of these students as there were previously, due primarily to the declining birth rate in the United States. Hence, universities will simply have a smaller pool to draw from in the 1980s than they had in the 1970s.

The situation grows even worse when you consider that it is widely believed that a smaller percentage of the students eligible for higher education will actually attend formal institutions. This is partially based on a projection of the existing situation, but other evidence is available too. We see a wise thrust to create and to encourage new kinds of institutions, some aimed at higher education, others aimed at more general markets.

Two studies conducted by the Carnegie Commission on Higher Education relate to this topic. One of them, Less Time, More Options: Education Beyond the High School, makes explicit recommendations in new directions. For example, one is that "professions, whenever possible, create alternate routes of entry other than full-time college attendance, and reduce attendance. . . ." But this report, done in 1971, still looks primarily toward the university. The 1973 report, Toward a Learning Society, proposes a wide range of alternate strategies, including such things as learning pavilions, private specialty schools, apprenticeship programs, and the more extensive use of libraries and museums. All of these are moves away from formal higher education as we know it today.

These new forms of education are partially dictated by the increasing need for providing a much wider range of continuing education than was needed in the past. These needs are a product of modern science and technology in that we now live in a society where change is much more rapid. No longer can a person expect to do the same thing during an entire lifetime. The nature of jobs will change as our society evolves. Buckminster Fuller, in mentioning this problem in Education Automation, states "We as an economic society are going to have to pay our whole population to go to school, and pay it to stay in school, as we are going to have to put our whole population into the educational process and get everybody realistically literate in many directions." Given this necessity, new modes of education for the continuing student will come into existence. These modes will have an impact on the traditional universities simply because they are available.

One sees various figures as to what the percentage drop will be in universities in the '80s. I am not an expert in such matters, but it seems that a 20% enrollment decline is at least a reasonable possibility. Whatever the drop, it will be drastic compared to our present situation. Any drop at all will immediately bring attention to the institutions of higher education.

This declining enrollment comes along with a phenomenon that is already well developed at the present time, increased public concern with education. This concern brings with it a remarkable change in attitude. At one time, education was seen as the magic door of

American Society, the way that a person could rise above the economic and social situation of the parents. Education was almost a panacea, recommended often as the solution to almost any problem that appeared in the society. We hardly need to do much reading in today's newspapers to see that attitudes toward education, both elementary and higher, have changed drastically in the last few years. Cities have closed schools for extensive periods by refusing to vote bond issues. Major universities have been hurt by legislatures. Governors have campaigned largely on the basis of cutting down the university budget and restricting it in various ways.

Not only is this political climate currently unfriendly and unfavorable to the existing educational institutions, but it is also one that seems to be continuing. It is often closely coupled with the view that these institutions are wasteful with money, so the most direct effect is budgetary. General economic problems are also likely in the 1980s; our current problems with energy, and our inability to solve them, almost guarantee that our country's economic system will have many crises in this period. Education will be particularly susceptible at such moments.

One of the most disturbing features in this situation of decreasing student population and unfavorable public attitude is how little attention these factors are receiving in our universities. Many administrators are aware of the possibilities, yet the vast majority of the faculty do not seem to be aware, or are only vaguely aware, of the likely consequences. There has been relatively little discussion on campuses of this coming crisis. I know of little long-range planning in institutions I am familiar with that is designed to respond to or cope with this problem.

Faculty who do hear of these possibilities often believe that it will all magically work out correctly in the end. They trust in the fact that universities have been around in their present form for quite a while, and that additional promotional efforts to attract students will work. Thus the topic is seldom discussed by academic bodies.

It is also important to mention, in discussing the future of higher education, some new forms of higher organizations that have come into prominence in recent years. Perhaps the most important of these, in terms of its implication for the future, is the Open University in Great Britain.

The form of the Open University is drastically different from those we are most familiar with. First, large amounts of time and money are put into curriculum development, an issue which is almost ignored in traditional American universities. Thus, the Open University may spend several hundred thousand dollars developing a course. Compare this with what happens when a faculty member prepares a course in a "standard" American university. The Open University also has very efficient delivery systems, mechanisms for student learning, that do <u>not</u> involve buildings, a large resident faculty, or other features that represent most of the cost of a conventional university. The results of the Open University experiment are positive in terms of the quality of education received and in terms of the cost effectiveness of this approach. Some features, such as the widespread use of television, are to some extent restricted to the

British environment, and American variants (such as the University of Mid-America) cannot yet claim the success of the Open University. Yet the structure does have great potential.

Given these factors, great pressures will exist to move toward more and more efficient forms of education. The computer will be in the right place at the right time to help meet these pressures. To explore some of the possibilities in the next section, I will examine how our present universities look from a student's point of view.

The University as Seen by Students

Most views of the university are presented from a faculty or administrative point of view. I would like to consider how the university appears to students.

The first thing to note is that a lot of things in present universities are not arranged for the student's convenience. The timing of classes, the question of who teaches what, the room assignments on campus, and the availability of the faculty are all arranged in ways to make them as convenient as possible for the instructor, but not necessarily convenient to the students. In this regard, we are comparable to some of the worst features of the medical profession, where the convenience of the doctor is clearly more important than the convenience of the patient.

Often coupled with this is an unfriendly attitude toward students. Many faculty are seen by students as unhelpful, sometimes even arrogant.

If we are concerned about preserving the university, even in a changed form, we must ask what it is offering the student at the present time. One way of thinking about this is to consider how the student spends time during a typical week. This will be different for different students, so any one choice will not reflect the full situation. The following table offers a possible breakdown in how a student, who is presumed to be working forty hours a week in taking four traditionally taught courses, possibly might spend time during a week:

40 hour/week total, 4 courses

Activity	Hours
Lecture	10
Lab	5
Direct contact with instructors	.5
Direct contact with TA's (tutorial)	1
Work with other students	3.5
Study of textbooks	11
Homework problems	9

The first thing to note about this table is that the lecture, typically thought of as the main teaching mode of the university, only takes about one-quarter of the total student time. Fully half of the

student's time, the twenty hours devoted to the textbook and homework, has nothing to do with the institution at all, and could be carried on quite independent of enrollment. There may be some feedback provided as to how well the student is doing in homework, but even this is vanishing in many schools at present. Some experimental evidence even indicates that this feedback may not be as valuable as the instructor feels.

Note that even an optimistic figure shows relatively little direct contact with the instructor. Although this feature is often stressed in university catalogs, all of us know realistically that, given large classes, the instructor can spend only a very small amount of time with each student.

Many of the ingredients are available, although in a somewhat different form, to the student in organizations such as the Open University, and they are often available in better form. That is, those courses offer a much more carefully designed set of study instructions to the student than typical university courses.

There is, however, one unique feature which the formal institution does offer the student. The "home" university cannot typically provide contacts with other students. This important learning tool is something that the conventionally organized one-location university can offer, and it may be perhaps the most important single contribution of the formal university. However, it never shows up on any of the usual judgements that the university makes about the quality of its own education, and typically would not be viewed as an important ingredient of the educational process.

New Structures for Universities

As the computer plays an increasingly larger role in the educational process, it is very likely that the organizational structures of universities will evolve for a given period of time. These changes will be much more dramatic than the actual role of the computer as a learning device.

Surprisingly little literature attempts to project ahead to possible futures for education. But there is an interesting project in George Leonard's Education and Ecstasy. Leonard pictures a school involving students from ages three to twelve. This school of the year 2000 shows an approach which will be possible with extensive use of educational technology to aid learning. Students have two essentially different kinds of learning environments, one for dealing with subject areas and one for developing interpersonal relationships. Only the first of these is heavily computer-dependent. This learning activity takes place in learning domes, with students working in front of large three-dimensional color displays, and with a full course management system knowing everything about the progress of each student.

Returning to higher education, we can see a number of possibilities for new class, curriculum, and organizational structures. Some aspects of these futures will be discussed in the next few sections.

Undoubtedly the first effects of computer technology to be felt will be changes in individual courses. These changes will only gradually begin to be perceived as general organizational and structural changes. Thus, the computer makes self-paced courses, whether in the PSI or other mastery-based traditions, more practical. We can expect increased use of computers in such courses, particularly in the large beginning courses that many faculty are not eager to teach.

Along with this increased use of self-paced courses will be the development within these new courses of better strategies for dealing with the nonverbal student, the student who is often unsuccessful in our current verbally oriented educational system. The use of computer graphics shows great promise in this area.

A natural extension of the idea of self-paced courses is self-paced curriculum. The same ideas which are applied to self-paced courses can be naturally applied to the entire curriculum. In the self-paced curriculum, no "courses" exist in any formal sense, no fixed discrete units that have to fit into some particular time limit. Rather, a student's progress would be on a continuous basis in several areas. This would imply that nothing like quarters or semesters are needed. With the computer-based record keeping, there is no reason why the student should not start a particular line of study at any time during the year and finish it at any time. (In many cases we would want to have built-in awards for prods to prevent this course period from dragging on too long, however). Under these circumstances some students would finish the typical fixed four-year education in two years, while others might take as long as six. Some would take modules that concentrate only in certain very narrow fields, but others might take, either by choice or by strictures of the curriculum, a much broader range of materials. The sequences will often be of a more practical nature than current ones, bridging the gap between theory and practice, building up student intuition for dealing with the situations that involve much practice.

Another current feature associated often with mastery learning is likely to be much accelerated by the increasing role of the computer. This is the shift from norm-based procedures, where students are compared with each other, to competency-based procedures, in which students perform according to standards.

Norm-based grading has been under a variety of attacks in the last fifteen years. Survey studies find very little correlation between students' grades and their adult achievement. Ohmer Milton, in Alternatives to the Traditional, quotes several such studies, including a 1965 analysis of forty-six comparison studies. In twenty-four of them, no relation was found between grade point averages and adult achievement. In nineteen, there was a slight relationship, and in only three was a higher relationship apparent. The writer of this report concludes that "present evidence strongly suggests that college grades bear little or no relationship to any measure of adult achievement."

Indeed, the situation may even be worse than this. A Berkeley committee in 1964, also quoted by Milton, concluded that "it seems reasonably clear that the grading system at all levels, including the graduate one, tends to award the conforming plodder and to penalize

the imaginative student who is likely to make a significant contribution in nearly any field." Any grading system based on the error curve should be viewed with suspicion; such a curve arises when the results are entirely random!

Competency-based testing is a natural adjunct of mastery-based learning. Objectives in each module are carefully specified, and the tests show explicitly that the students have or have not mastered these competencies. This is not to say that competency-based grading will necessarily lead to better correlation with adult achievement. But at least there is a possibility, and at least in this type of grading, the student will have a clearer understanding of what further study is needed.

The self-paced curriculum will probably also lead to entirely different views of how "credit" is to be assigned to student achievement, and how eventual certification of something like a degree is to be attained. I leave the details to your imagination.

One of the most serious threats to conventional universities that choose not to adapt to the new technology will be the increased availability of alternate modes of education, a topic already mentioned in several of the Carnegie Commision reports quoted earlier. We can expect that learning activities will, to a considerable extent, migrate out of the formal institutions and go into public environments or into the home environment. Much more of this will be learning at a distance, in the sense of the open universities. Public libraries, and other civic centers, provide an ideal location for stand-alone computer systems, with a wide variety of learning material available from the reserve desk of the library. Similarly, the learning pavilions suggested for adult education will play an important role if structures of this kind do develop. It seems likely that the most realistic possibility is that the learning pavilions will be developed in the context of the public library.

As personal computers become more widely available in the home environment, and as these computers increase in sophistication and decrease in cost, it is almost certain that a good bit of learning will take place in the home, not in the public environment at all. I would not want to claim, as some of my colleagues perhaps would, that <u>all</u> learning would take place in the home. Some of the social aspects of learning, the interaction with other people either for direct learning purposes or for other purposes, is important to what goes on in most of our universities today. Hence, I would not see a situation with <u>no</u> contact between students. But much of the learning process, mediated by technology, could take place at a distance. Note that the self-paced curriculum, implemented with extensive use of computers and other media, fits in very well with the general ideas of learning at a distance.

All these changes imply, within the institutions of higher education that continue to exist, a new role for the faculty, different from that of faculty in the present university. The faculty, as we have noted, conceives of itself as the main bearer of learning. Yet the student sees the faculty as contributing only a small quantity to the learning that goes on. Learning is fundamentally a very personal activity, and can only be done in the full sense by the individual.

The faculty may still serve as a role model for many students. Even more importantly, we can expect the faculty to be a solver of learning problems which are not solvable by all the other learning possibilities, computers and otherwise. Thus, the faculty will spend far more time working with individual students, particularly students who are departing seriously from the average (either above or below), where average now indicates educational progress. The main beneficiary will be the weak student, the student who cannot survive in our present large universities. Not all faculty will feel happy with this role, but for many faculty it will seem superior to be able to deal individually with students who can best benefit from faculty attention.

We can also expect the institutional structure of the university to change, although it is not unique to the university. The features of this new structure have been described by Peter Drucker, in such books as Technology, Management, and Society. He argues that the traditional pyramidal structure will be supplanted by a more complex-free form of organizational structure, with a great deal more autonomy of the components, but with special attention to careful setting of objectives and communication between the components.

Organizational Change

Many people associated with universities, hearing the earlier sections of this talk, may be convinced that my ideas sound unrealistic. Universities are very conservative. History shows that they are very resistant to changes of all kinds.

The kinds of changes which are "approved" within higher education are often meaningless changes. For example, we see a constant vacillation between quarters and semesters, and we see changes in the grading system, such as allowing or not allowing plus and minus grades. Somehow these changes are "safe" enough to be considered as changes. Faculty committees are often willing to devote huge amounts of time to debating them, in spite of the fact that the empirical evidence very strongly indicates that they represent no real change at all. If quarters were clearly superior, this oscillation would not take place. But the people in power in universities, faculty and administration, are the successful university products. They tend to argue, at least implicitly, that if the university produced them, it must be good for the current students. The methods of the university work for these people, and they are the decision makers in the next generation.

Typically the kinds of changes which occur easily within organizational structures are changes which do not threaten the organization as a whole, and which fit in well with the existing pattern. But the present situation is not a "normal" one. What is being described is not in any sense a normal evolution from within. Rather, it is a revolution. Like most revolutions, it will be pushed primarily by factors outside the university. Rational ways of viewing what "should" happen ("rational" as seen by many university professors) will have little effect on the situation. Under these circumstances, with very strong external factors driving the situation, the usual mechanisms for slowing internal change are likely to be ineffective. Change will be imposed from outside.

Sometimes the rapid evolution will be demanded by state legislatures, regents, or even by public laws. While it does not apply directly to four-year institutions in California, the proposed Jarvis-Gann amendment in California, Proposition 13 on the ballot this fall, is a sign of the type of pressure which can be applied. It does no good to argue that public schools and community colleges will only evolve gradually. If the Jarvis amendment passes, it appears from the evidence currently on hand that drastic changes will occur.

Universities will not be immune to similar approaches in the 1980s. Indeed, given changing public attitudes toward higher education, and given the real possibilities of cycles of economic troubles, our institutions of higher education will almost certainly be forced to react to these external pressures. If choice is available, some will choose to and some will not. The consequences will be drastic for those that choose not to.

How we can prepare for these changes? First, it seems critical to alert faculty and administrators that these changes are at least possible and perhaps likely. Leaving faculty with the view that somehow they will muddle through, and everything will be pretty much as it was before, is morally unacceptable at the present time.

Within university administrations there needs to be some conscious effort to plan toward the future, not just toward the simple handling of day-to-day current crises. Very little such planning is currently visible.

Perhaps our greatest need at the present time is for effective educational leadership. It is often said that current administrators in colleges and universities are seldom leaders in any positive sense. Many seem beleaguered, just trying to keep the ship afloat. Others are interested in power, moving toward more and more powerful positions, seeing the university as a power structure. Positive leadership is rare. We don't require individual strong men, but leaders capable of moving others and capable of challenging entrenched interests.

We badly need those educational leaders, vigorous individuals with forward views, who will see in the unpleasant circumstances new forms and structures for the university, and will work toward creating them. Our universities can come out of this entire process as more interesting institutions, as better institutions, than they are at present. But it will take leadership.